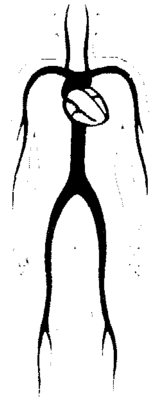
CR-160609

SΦT



```
(NASA-CR-160609)  CARDIOPULMONARY DATA                    N80-33084
ACQUISITION SYSTEM.  VERSION 2.0, VOLUME 2:
DETAILED SOFTWARE/HARDWARE DOCUMENTATION
Final Report (Technology, Inc., Houston,                  Unclas
Tex.)  195 p HC A09/MF A01          CSCL 05B G3/52         33201
```

TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION

NSC-18783

FINAL REPORT

ON THE

CARDIOPULMONARY DATA ACQUISITION SYSTEM

VERSION 2.0

VOLUME 2

DETAILED SOFTWARE/HARDWARE DOCUMENTATION

Prepared for the NASA Johnson Space Center
Cardiovascular Research Laboratory

February 18, 1980

P. O. BOX 58827 • HOUSTON, TEXAS 77058

FINAL REPORT

ON THE

CARDIOPULMONARY DATA ACQUISITION SYSTEM

VERSION 2.0

VOLUME 2

DETAILED SOFTWARE/HARDWARE DOCUMENTATION

Prepared for the NASA Johnson Space Center
Cardiovascular Research Laboratory

February 18, 1980

Contract NAS9-14880
Project 0160-20

TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION
16821 Buccaneer, Suite 206
Houston, Texas 77058

Table of Contents

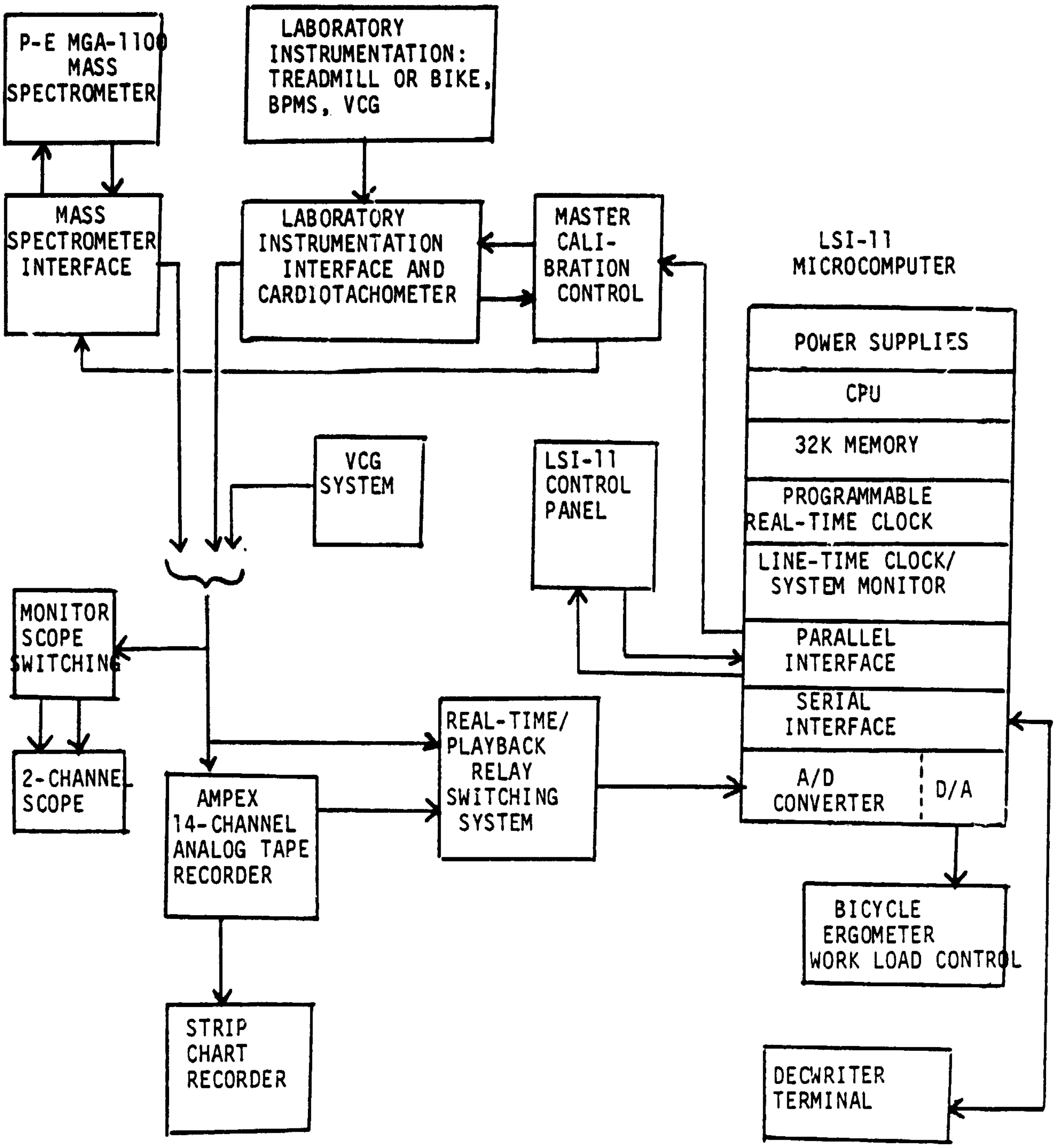
Volume I (User's Guide) Page

Volume II (Detailed Software/Hardware Documentation)

APPENDIX F

WIRING AND TIMING DIAGRAMS

CDAS - CARDIOPULMONARY LABORATORY
SIMPLIFIED HARDWARE CONFIGURATION


P-E MGA-1100 MASS SPECTROMETER
LABORATORY INSTRUMENTATION: TREADMILL OR BIKE, BPMS, VCG
MASS SPECTROMETER INTERFACE
LABORATORY INSTRUMENTATION INTERFACE AND CARDIOTACHOMETER
MASTER CALI-BRATION CONTROL
LSI-11 MICROCOMPUTER
POWER SUPPLIES
CPU
32K MEMORY
PROGRAMMABLE REAL-TIME CLOCK
LINE-TIME CLOCK/ SYSTEM MONITOR
PARALLEL INTERFACE
SERIAL INTERFACE
A/D CONVERTER
D/A
VCG SYSTEM
LSI-11 CONTROL PANEL
MONITOR SCOPE SWITCHING
2-CHANNEL SCOPE
REAL-TIME/ PLAYBACK RELAY SWITCHING SYSTEM
AMPEX 14-CHANNEL ANALOG TAPE RECORDER
BICYCLE ERGOMETER WORK LOAD CONTROL
STRIP CHART RECORDER
DECWRITER TERMINAL

JSC CARDIOPULMONARY LABOPATORY
STRESS TEST TAPE LOG FORM

Analog Tape #______ Subject Name:______________________________ Subject #______________

Date:______________ DOY:________ SS#______________ AGE:______ HT.(cm)______ WT.(kg)______

CALIBRATION DATA

Analog Tape Channel	Data	Calib. Value (low/high)
3	VCG-X	1 mV
5	VCG-Y	
7	VCG-Z	
2	HEART RATE	60/180 bpm
4	ELEVATION	8/32%
	WORK LOAD	60/240 watts
6	SPIRO. VOLUME	0/8 liters
9	TREAD. SPEED	2/8 mph
	BIKE SPEED	20/80 rpm
11	SYSTOLIC BP	50/200mmHg
13	DIASTOLIC BP	50/200mmHg
10	O_2	0/______%
14	N_2	0/______%
12	CO_2	0/______%
1	TAPE REF. (Direct)	
8	IRIG-B TIME (Direct)	

CALIBRATION START TIMES:

ECG: ____________________

Low: ____________________

High: ____________________

Ambient Air: ____________________

AMBIENT CONDITIONS:

O_2: ____________________ %

N_2: ____________________ %

CO_2: ____________________ %

Temperature: ______________ ° C

Bar. Pres.:______________ mmHg

TEST TYPE (Circle One) Bicycle Ergometer Treadmill (Modified Balke) Treadmill (Modified Bruce)

Begin Rest: ____________________(10 min) Begin Exercise: ____________________

Begin Recovery: ____________________(8 min) Begin Recovery: ____________________

Comments/Anomalous Conditions: ______________________________

Revised Feb. 1980

CDAS INPUT/OUTPUT CONNECTIONS

VCG, BPMS, Front Panel, and Analog Interface

BPMS SBP analog output → to analog interface SBP input
BPMS DBP analog output → to analog interface DBP input
BPMS Korothoff Sounds output → to analog interface K-Sounds input
BPMS Pressure analog output → (not used)

Spirometer controller output → to analog interface spir. vol. input

---

Treadmill speed & elevation outputs }
Bicycle ergometer speed & WL outputs } → to front panel treadmill/bicycle switch

---

Treadmill/Bicycle switch outputs → to analog interface speed & work load/ elevation inputs

---

LSI-11 D/A outputs (0 or 1) → to ergometer work load control input

---

VCG system X lead output }
VCG Y lead output } → to front panel VCG switch
VCG Z lead output }

Front panel VCG trigger select switch output → to ECG trigger inputs of BPMS and cardiotachometer

---

Cardiotachometer output → to analog interface HR input and to front panel DPM switch

---

All analog interface outputs }
Front panel BNC DPM/Scope Inputs } → to DPM/Scope Monitor switches

---

DPM/Scope Monitor switches outputs → to front panel DPM select switch and oscilloscope vertical inputs
Front panel DPM swtich outputs → to DPM

All analog interface outputs }
Mass spectrometer interface outputs } → to patch panel (see Chart)

T.C. generator IRIG-B code out → to analog tape chan. 8 record input
Tape recorder chan. 8 reprod. output → to T.C. generator/reader code in

---

Abbreviations:

VCG	Vectorcardiograph
BPMS	Blood Pressure Measuring System (Sphygmometrograph)
WL	Work Load (of bicycle ergometer)
D/A	Digital to Analog output from LSI-11
T.C.	Time Code
DPM	Digital Panel Meter

CDAS Relay Patch Panel, Analog Tape Recorder, LSI-11 A/D Inputs

Signal Name	Signal Source	Patch Panel Relay Pole No.	Tape Rec. Playback Ch. (for Relay Input #2)	Relay Output to Butter Amp. Input #	Buffer Amp. Output to A/D Input #	Strip Chart Chan. No. (for Tape Output)
Spir. Vol.	Spiro. Contr.	0	6	10	0	5
O_2	Mass	1	10	1	1	
N_2	Spectrometer	2	14	2	2	
CO_2	Interface	3	12	3	3	
Ht. Rate		4	2	4	4	
WL/Elev.	Analog	5	4	5	5	6
Speed	Interface	6	9	6	6	7
SBP	Outputs	7	11	7	7	
DBP		8	13	8	8	
(Spare)		9				
(Spare)		10				
24v for RT/PB lights	Pwr. Supply	11				
Tape Spd.Ref.	Tape Rec.		1*			
VCG-X	VCG	not	3*			1
VCG-Y	System	switched	5*			2
VCG-Z			7*			3
TRIG-B Time Code	Time Code Generator		8*			8**

NOTES:

(1) Except for patch panel relay pole #11:
- All relay #1inputs are from outputs of analog and mass spectrometer interfaces
- All relay #2 inputs are from analog tape recorder playback (reproduce) outputs.

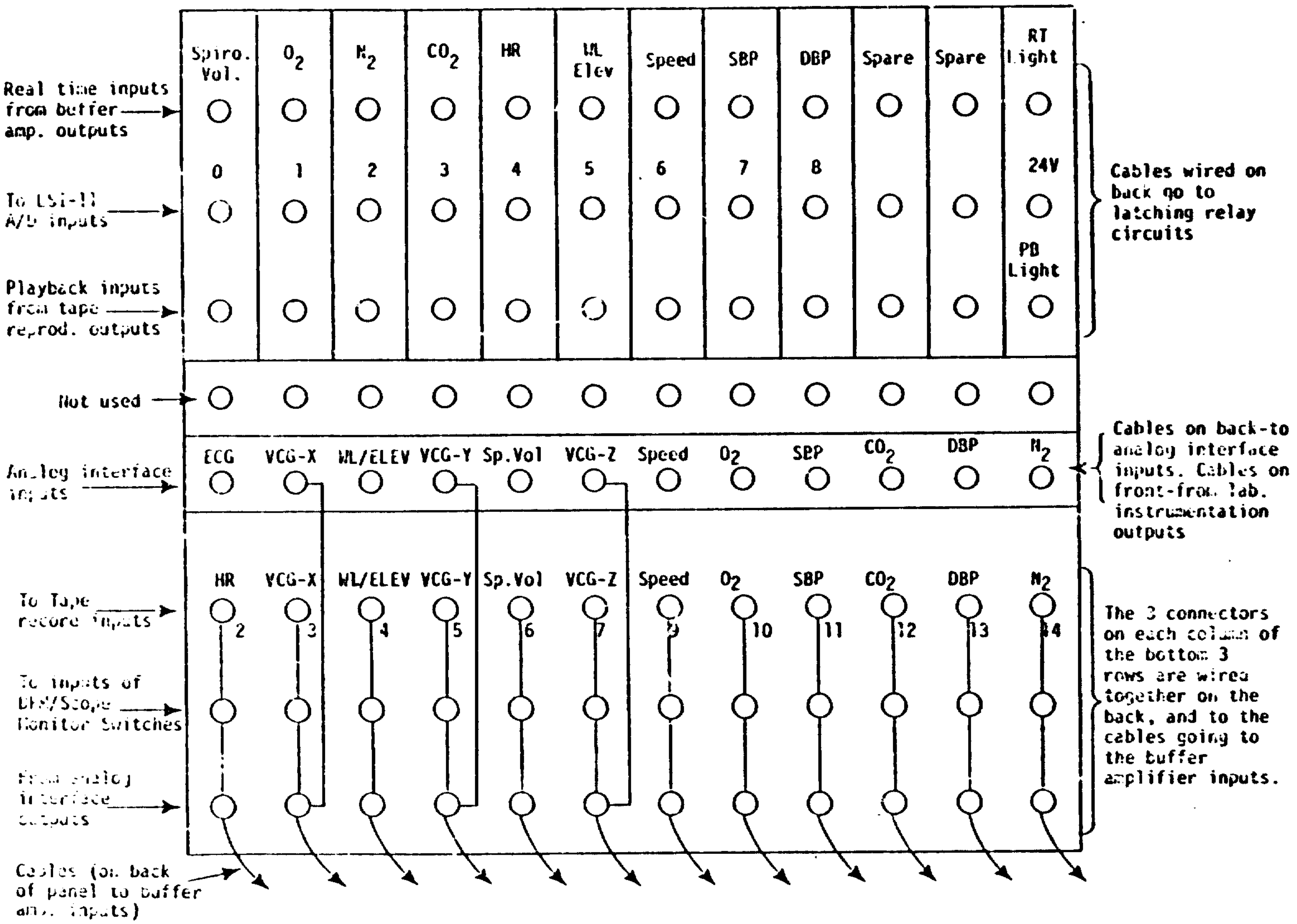
(2) Patch panel relay pole #11 is used to control the Real Time-Playback lights, which indicate the status of the latching relay. The pole of the switch is connected to the 24v. supply, the #1 contact goes to the Real Time light, and the #2 contact goes to the Playback light.

(3) All cable shields (signal returns) are switched along with the cable center conductors, and all isolated from each other at the patch panel.

(4)* These signals are not switched by the patch panel relay.

(5)** Slow code, not IRIG-B, is sent to strip chart recorder channel 8.

CDAS LATCHING RELAY PATCH PANEL


Spiro. Vol.
O2
N2
CO2
HR
WL Elev
Speed
SBP
DBP
Spare
Spare
RT Light
Real time inputs from buffer amp. outputs
0
1
2
3
4
5
6
7
8
24V
To LSI-11 A/D inputs
PB Light
Playback inputs from tape reprod. outputs
Cables wired on back go to latching relay circuits
Not used
ECG
VCG-X
WL/ELEV
VCG-Y
Sp.Vol
VCG-Z
Speed
O2
SBP
CO2
DBP
N2
Analog interface inputs
Cables on back-to analog interface inputs. Cables on front-from lab. instrumentation outputs
HR
VCG-X
WL/ELEV
VCG-Y
Sp.Vol
VCG-Z
Speed
O2
SBP
CO2
DBP
N2
To Tape record inputs
2
3
4
5
6
7
9
10
11
12
13
14
To inputs of BPMS/Scope Monitor Switches
From analog interface outputs
The 3 connectors on each column of the bottom 3 rows are wired together on the back, and to the cables going to the buffer amplifier inputs.
Cables (on back of panel to buffer amp. inputs)

Notes:

(1) Connectors on row 2 are switched to corresponding connectors on row 1 (top) in Real Time mode.
(2) Connectors on row 2 are switched to corresponding connectors on row 3 in Playback mode.
(3) Corresponding connectors on rows 6, 7, and 8 are wired together and to coax cables on back of panel.
(4) "Back" of panel faces inside of rack; "front" of panel faces the rear door of rack.
(5) Signal in "ECG" connector comes from front panel X-Y-Z switch, and goes to inputs of cardiotachometer and BPMS.

LSI-11 COMPUTER CONTROL PANEL AND INTERFACE CABLES

CDAS MICROCOMPUTER HARDWARE REQUIREMENTS

Supplier	Model No.	Description
MDB	MLSI-KD11-HA	LSI-11 CPU
MDB	MLSI-KEV11	Extended Arithmetic Option
MDB	MLSI-MSV11-DD	32K x 16 bit RAM, on-board refresh
MDB	MLSI-DRV11C	Parallel Interface
MDB	MLSI-SMU	System Monitoring Unit/Line Time Clock/ Front Panel Controls/Bus Terminator
MDB	MLSI-BPA84	Backplace - Card Guide Assembly
MDB	MLSI-BA11-000	Enclosure
MDB	MDB-250-T-5/12	Power Supply
MDB	MLSI-DLV11	Asynchronous Interface
MDB	MLSI-KW11P	Programmable Real-Time Clock
ADAC	1030-16PD-A-3PGA-2-A-P	Analog Data Acquisition and Control System
Data Systems	DSD 210-L11-2	Dual Flexible Disk System
DEC	LA36	Decwriter II Hard Copy Terminal

CVDAS Input/Output Signals - Parallel Interface

Input Register - Address 167774

Input Register Lines	Function
IN 00	CALIB. Switch
IN 01	REST Sw.
IN 02	EXERCISE Sw.
IN 03	RECOVERY Sw.
IN 04	PAUSE Sw.
IN 05	END TEST Sw.
IN 06	FVC Sw.
IN 07	(Spare Switch)
IN 08	not used
IN 09	
IN 10	
IN 11	
IN 12	
IN 13	
IN 14	
IN 15	

Output Register - Address 167772

Output Register Lines	Function
OUT 00	CALIB. Light
OUT 01	REST Light
OUT 02	EXERCISE Light
OUT 03	RECOVERY Light
OUT 04	PAUSE Light
OUT 05	END TEST Light
OUT 06	FVC Light
OUT 07	(Spare Light)
OUT 08*	TEST TYPE 0
OUT 09*	TEST TYPE 1
OUT 10	Decr. TM Elev.
OUT 11	Incr. TM Elev.
OUT 12	Incr. TM Speed
OUT 13	Decr. TM Speed
OUT 14	not used
OUT 15	not used

*See below

Control/Status Register - Address 167770
Control Lines (used as outputs)

CSR 1	CSR 0	Operating Mode
0	0	Normal (Operate)
0	1	Low Calib.
1	0	High Calib.
1	1	Ambient Air/Standby

Test Type Control Lines

OUT 09	OUT 08	Selected Test Type
0	0	Treadmill
0	1	Bicycle Ergometer
1	0	LBNP
1	1	Other

LSI-11 SERIAL INTERFACE CABLE

Computer 40-Pin Board Connector		Agile A1 Terminal 25-Pin EIA Connector	
Signal	Pin No.	Signal	Pin No.
——		Protective Ground	1
EIA Received Data In	8	Transmitted Data	2
Transmitted Data	6	Received Data	3
Signal Ground	1,2	Signal Ground	7
Data Set Ready	22	Data Terminal Ready	20
Data Terminal Ready	26	Carrier Detect	8

LSI-II ANALOG/DIGITAL CONVERTER CABLE

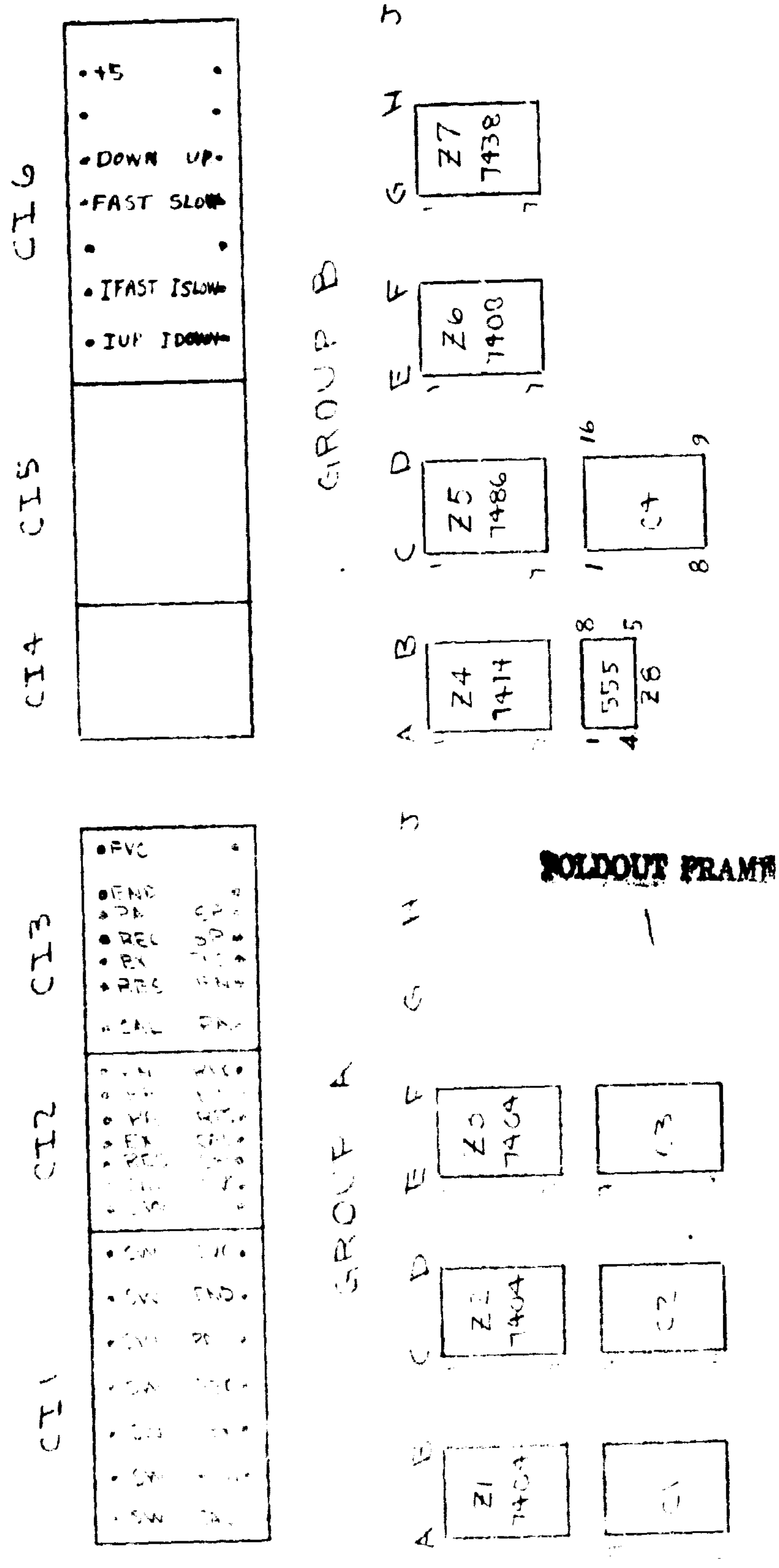
Signal Name	A/D Board Connector C1 Pin No.	Analog Cable 50-Pin Connector J1 Pin No.	Coax Cable BNC Connector Designation
A/D Ch. 0 in	31	16	0
A/D Ch. 1 in	29	15	1
A/D Ch. 2 in	27	14	2
A/D Ch. 3 in	25	13	3
A/D Ch. 4 in	23	12	4
A/D Ch. 5 in	21	11	5
A/D Ch. 6 in	19	10	6
A/D Ch. 7 in	17	9	7
A/D Ch. 8 in	32	33	8
A/D Ch. 9 in	30	32	9
A/D Ch. 10 in	28	31	10
A/D Ch. 11 in	26	30	11
A/D Ch. 12 in	24	29	12
A/D Ch. 13 in	22	28	13
A/D Ch. 14 in	20	27	14
A/D Ch. 15 in	18	26	15
D/A Ch. 1 out	37	46	D1
D/A Ch. 1 ret.	38	45	(D1 Shield)
D/A Ch. 2 out	39	44	D2
D/A Ch. 2 ret.	40	43	(D2 Shield)
Amp. Lo in (Source Ret.)	33	50	(A/D Input Shields)
Power Return (Chassis Grd.)	34	49	N.C.

Related Equipment

Seperate reports are available on the following items, which are used with the LSI-11 CVDAS system for tests in the JSC Cardiovascular Laboratory:

Microprocessor - Based Cardiotachometer

Analog Interface and Control System for the
JSC Cardiovascular Laboratory

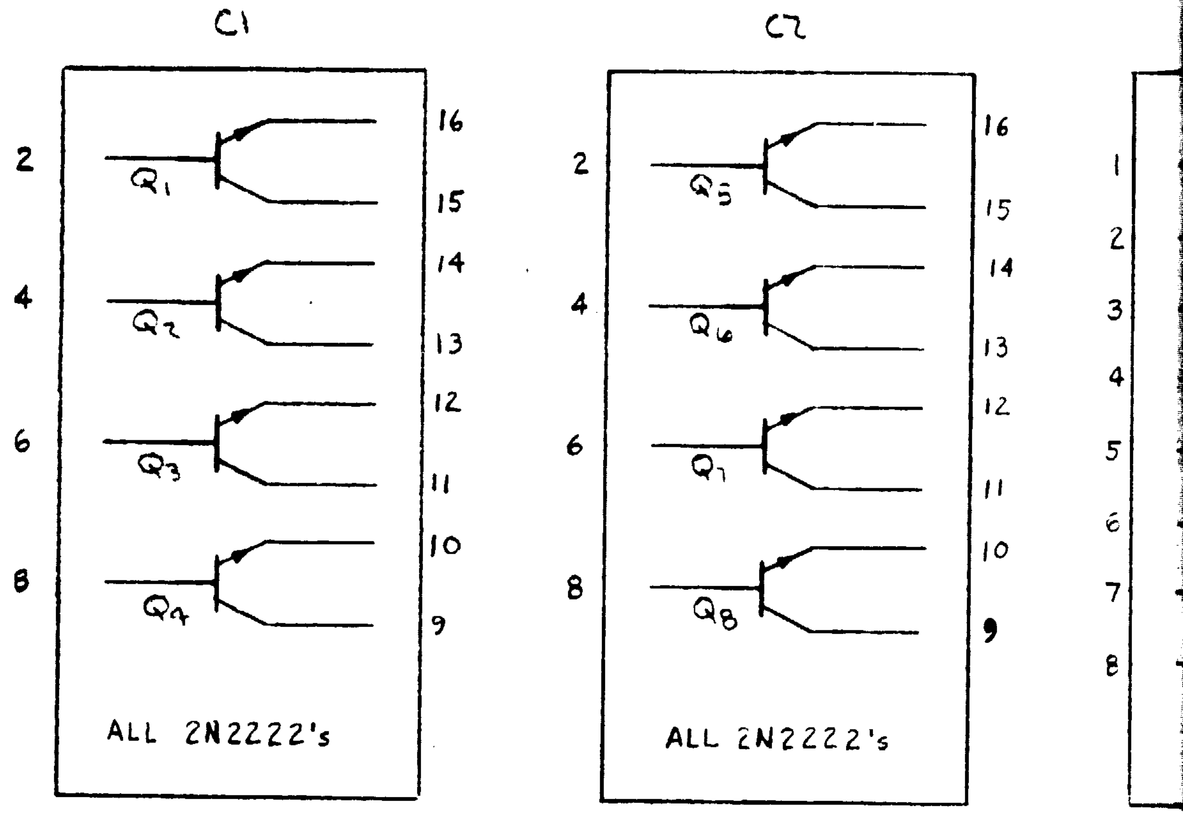


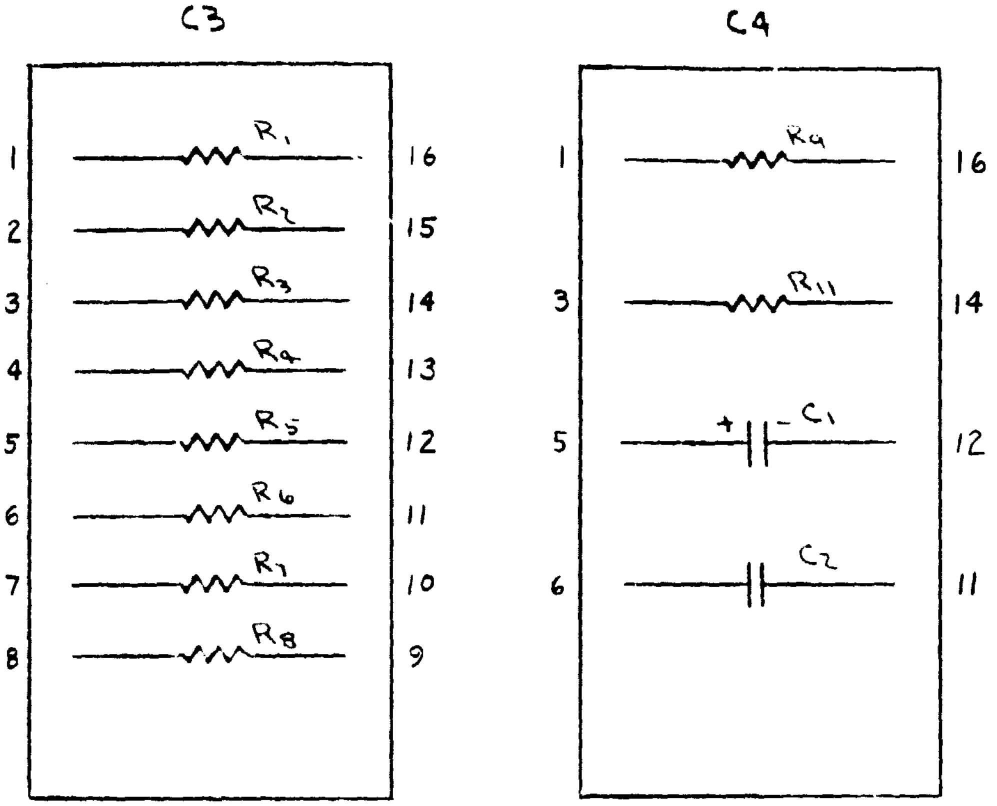
FOLDOUT FRAME

1

PRECEDING PAGE BLANK NOT FILMED

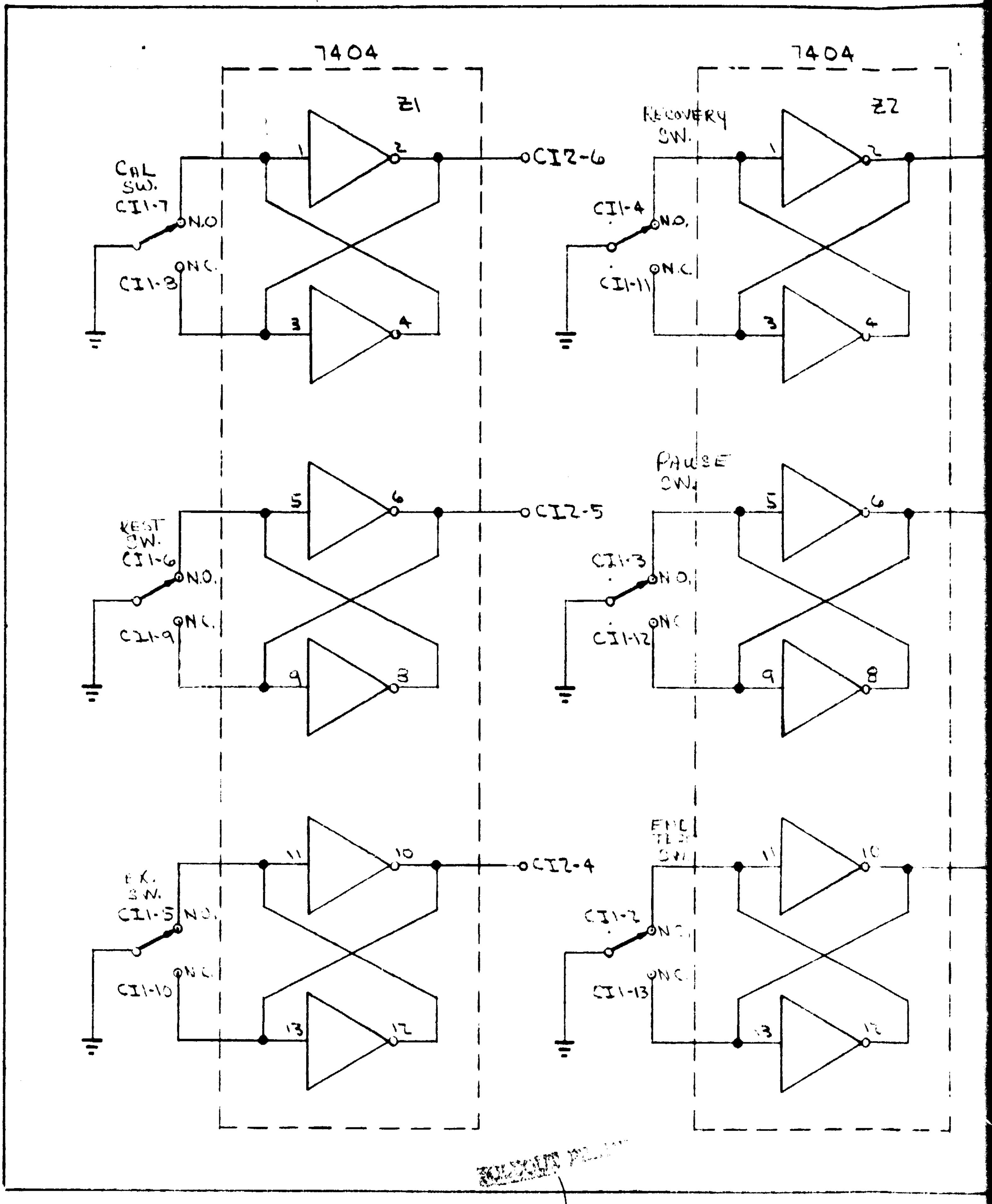
		TECHNOLOGY INCORPORATED LIFE SCIENCES DIVISION [illegible]
DRAWN BY M.H. UTLEY	DATE 10-16-78	TITLE CVDAS - CONTROL PANEL CIRCUIT [illegible] [illegible]
DESIGN ENG		
PROJ ENG		DWG NO. TH7942-1A01
		SHEET 1 [illegible]

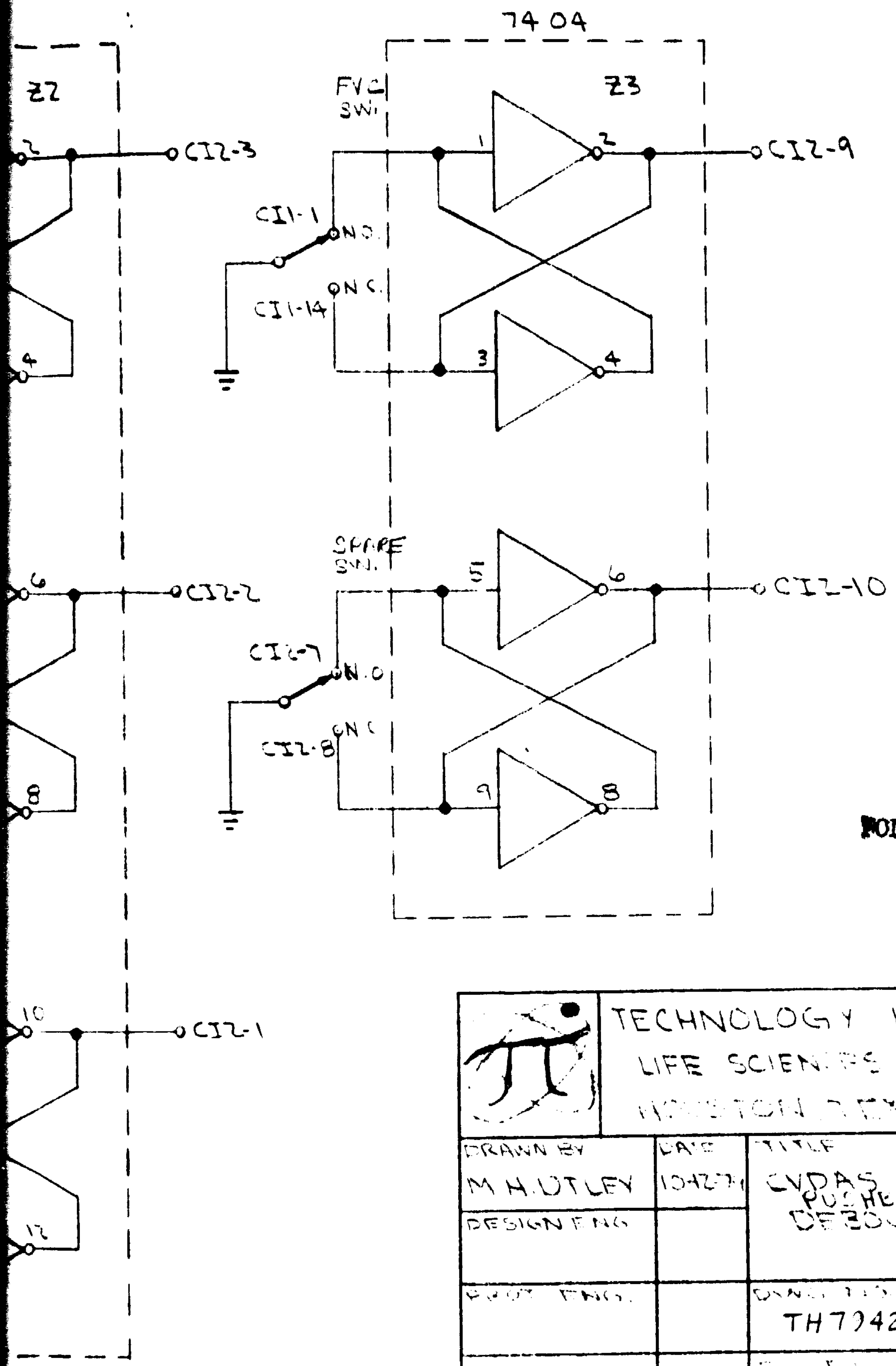

C1
2
4
6
8
Q1
Q2
Q3
Q4
16
15
14
13
12
11
10
9
ALL 2N2222's
C2
2
4
6
8
Q5
Q6
Q7
Q8
16
15
14
13
12
11
10
9
ALL 2N2222's
1
2
3
4
5
6
7
8



	TECHNOLOGY INCORPORATED LIFE SCIENCES DIVISION HOUSTON, TEXAS 77058

DRAWN BY M.H. UTLEY	DATE 10-25-79	TITLE CVDAS - CONTROL PANEL COMPONENT LAYOUT
DESIGN ENG.		
PROJ ENG.		DWG NO TH7942-1A02
		SHEET 1 OF 1

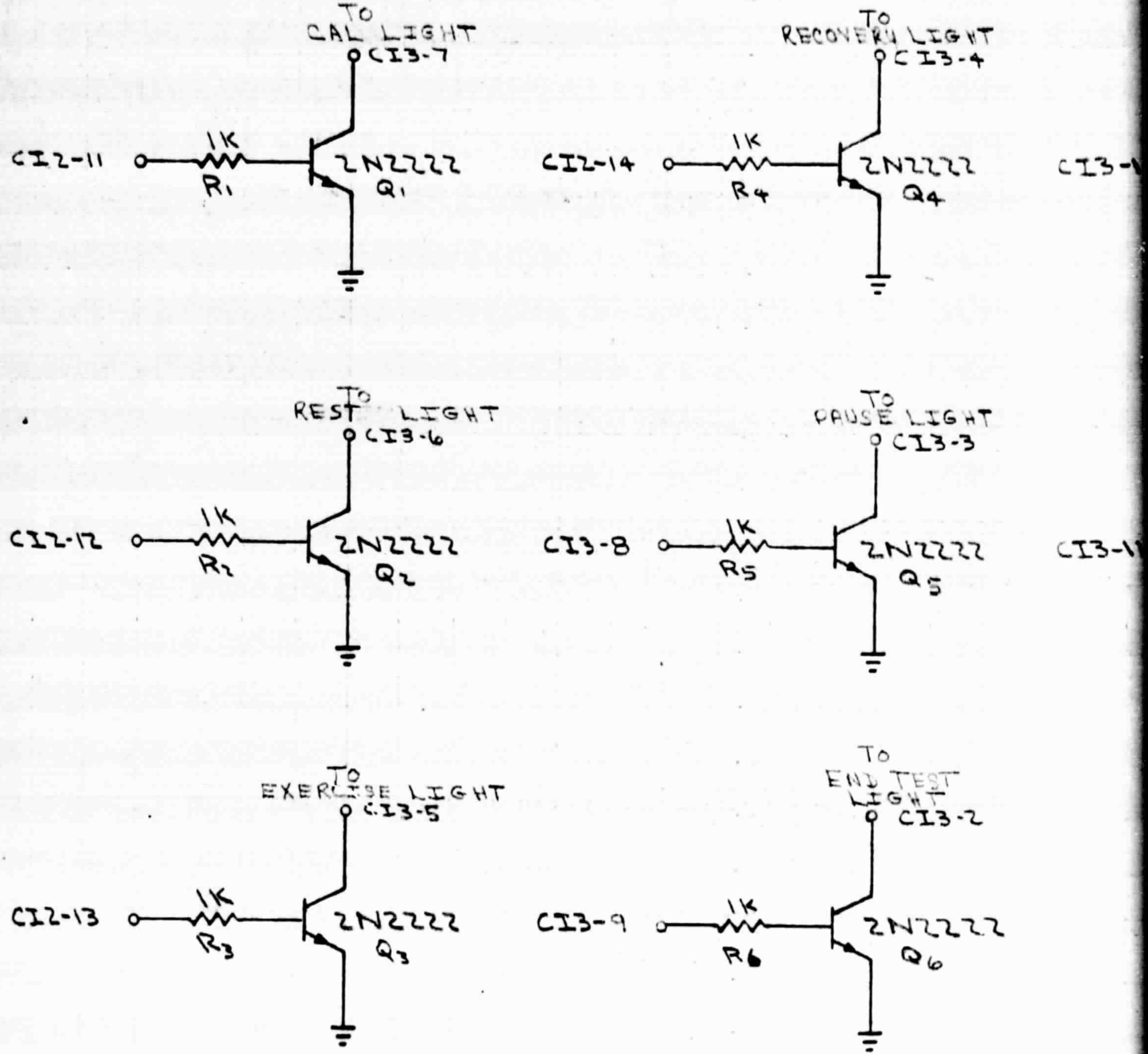

7404
Z1
7404
Z2
CAL SW. CI1-7
N.O
N.C.
CI1-3
1
2
3
4
CI2-6
RECOVERY SW.
CI1-4
N.O.
N.C.
CI1-11
REST SW. CI1-6
N.O.
N.C.
CI1-9
5
6
9
8
CI2-5
PAUSE SW.
CI1-3
N.O.
N.C
CI1-12
EX. SW. CI1-5
N.O.
N.C.
CI1-10
11
10
13
12
CI2-4
END TEST SW.
CI1-2
N.C.
N.C.
CI1-13



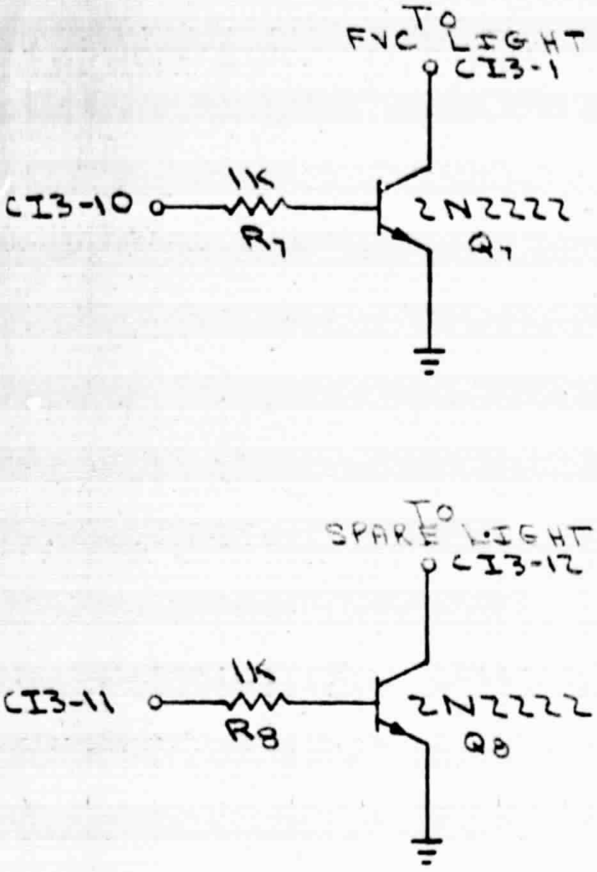
FOLDOUT FRAME

2

	TECHNOLOGY INCORPATED LIFE SCIENCES DIVISION HOUSTON TEXAS	
DRAWN BY M H. UTLEY	DATE 10-12-74	TITLE CVDAS CONTROL PANEL PUSHBUTTON SWITCH DEBOUNCE CIRCUIT
DESIGN ENG.		
PROJ. ENG.		DWG. NO. TH7942-1E01
		SHEET 1 OF 1



FOLDOUT FRAME

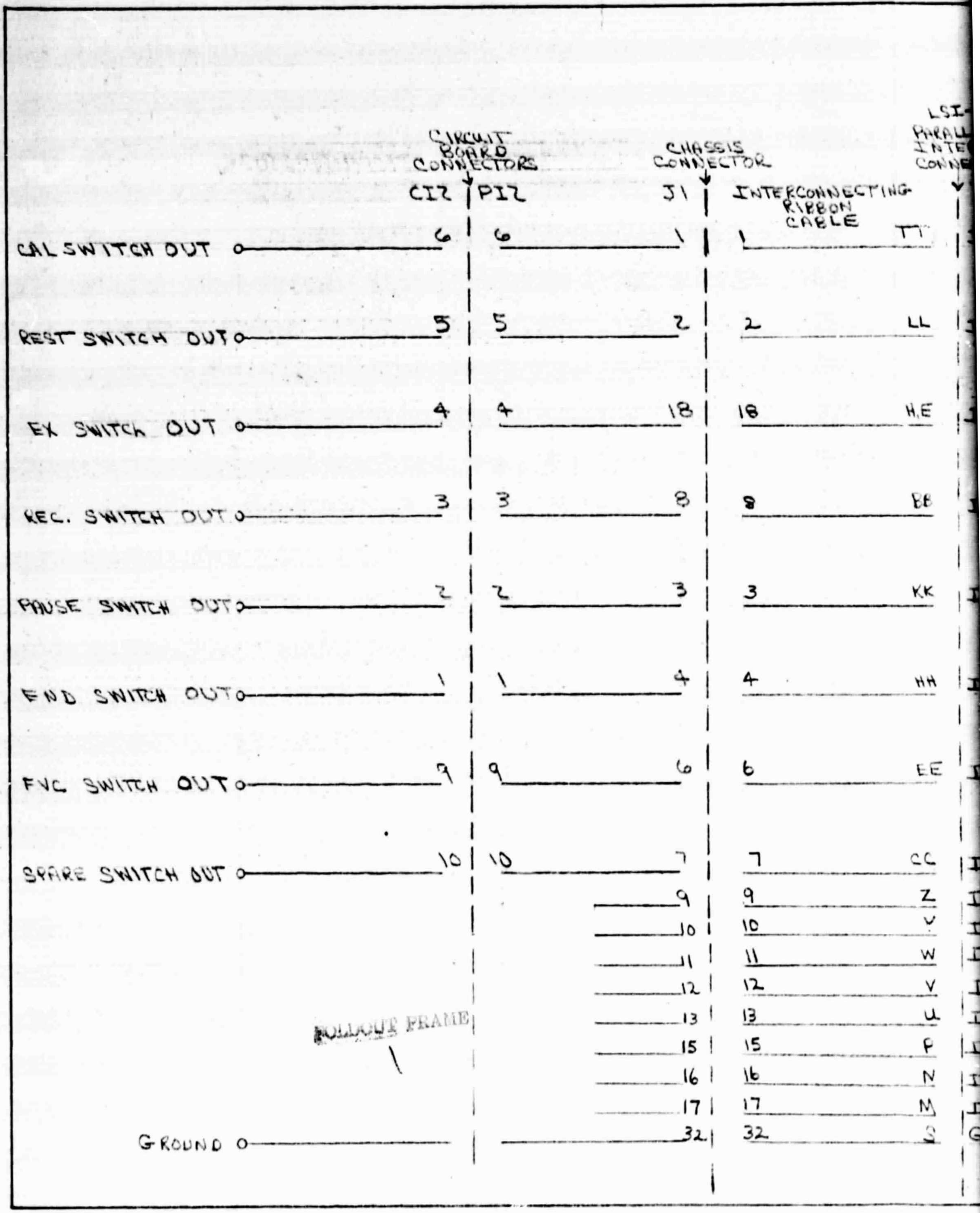


FOLDOUT FRAME

2

NOTES: (1) COMMON SIDE OF ALL LIGHTS IS CONNECTED TO +24V SUPPLY OUTPUT.

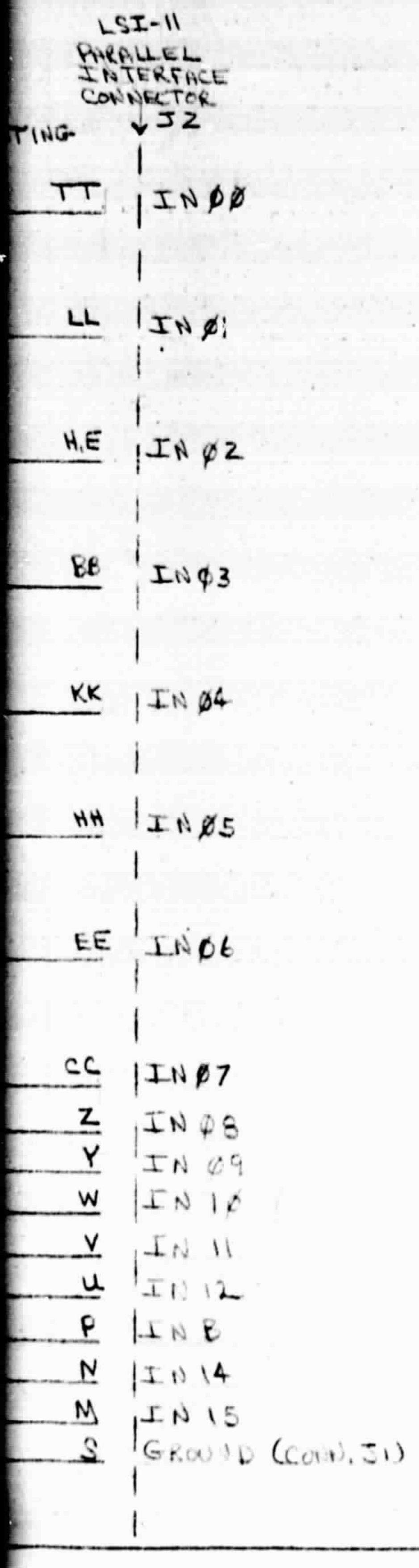
TECHNOLOGY INCORPORATED LIFE SCIENCES DIVISION HOUSTON, TEXAS 77058		
DRAWN BY M. H. UTLEY	DATE 10-25-79	TITLE CVDAS- CONTROL PANEL PUSHBUTTON LIGHT DRIVERS
DESIGN ENG		
PROJ. ENG		DWG. NO. TH7942-1E02
		SHEET 1 OF 1



	CI2	PI2	J1	INTERCONNECTING RIBBON CABLE	
CAL SWITCH OUT	6	6	1	1	TT
REST SWITCH OUT	5	5	2	2	LL
EX SWITCH OUT	4	4	18	18	H,E
REC. SWITCH OUT	3	3	8	8	BB
PAUSE SWITCH OUT	2	2	3	3	KK
END SWITCH OUT	1	1	4	4	HH
FNC SWITCH OUT	9	9	6	6	EE
SPARE SWITCH OUT	10	10	7	7	CC
			9	9	Z
			10	10	Y
			11	11	W
			12	12	V
			13	13	U
			15	15	P
			16	16	N
			17	17	M
GROUND			32	32	S

FOLDOUT FRAME

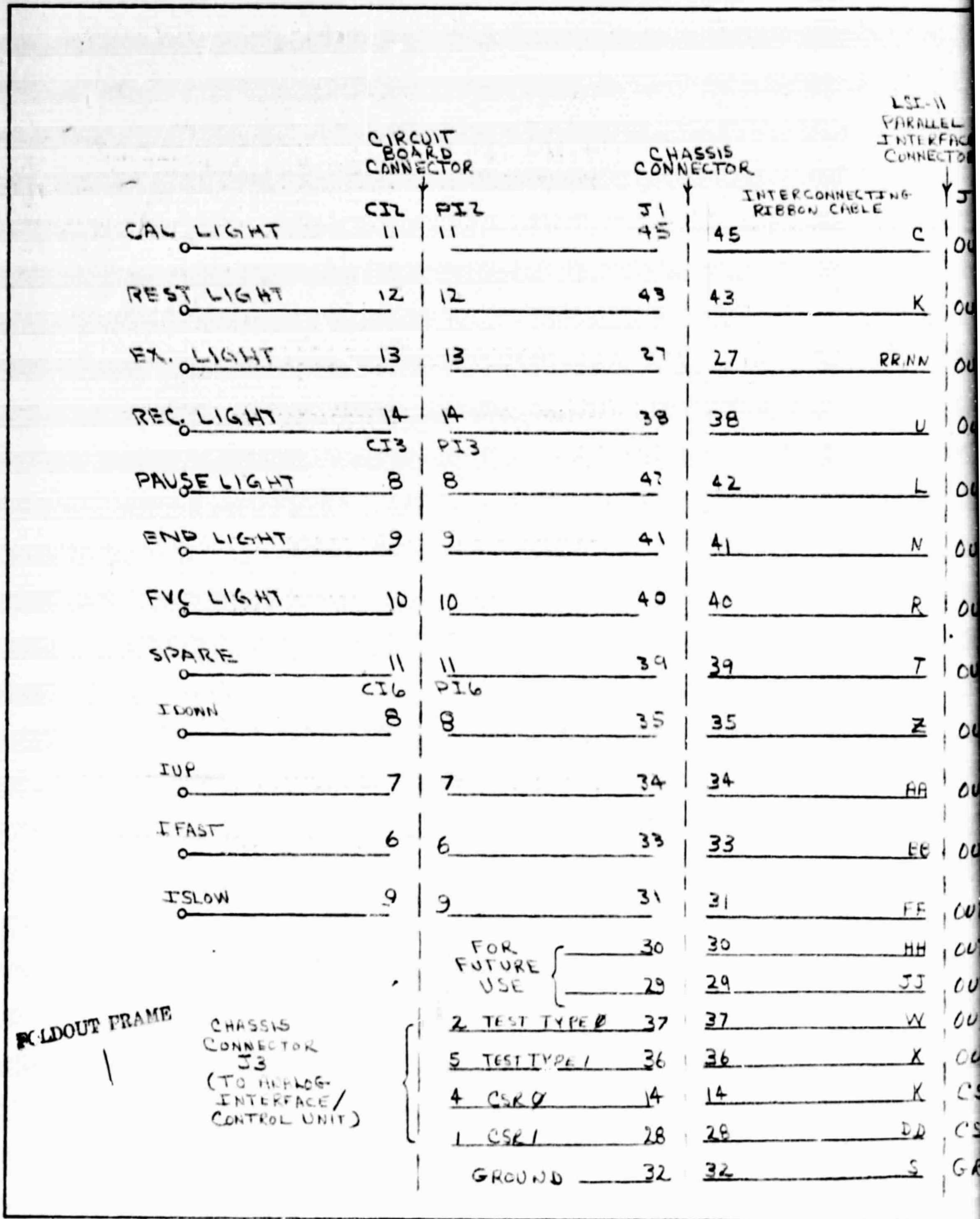
1



FOLDOUT FRAME 2

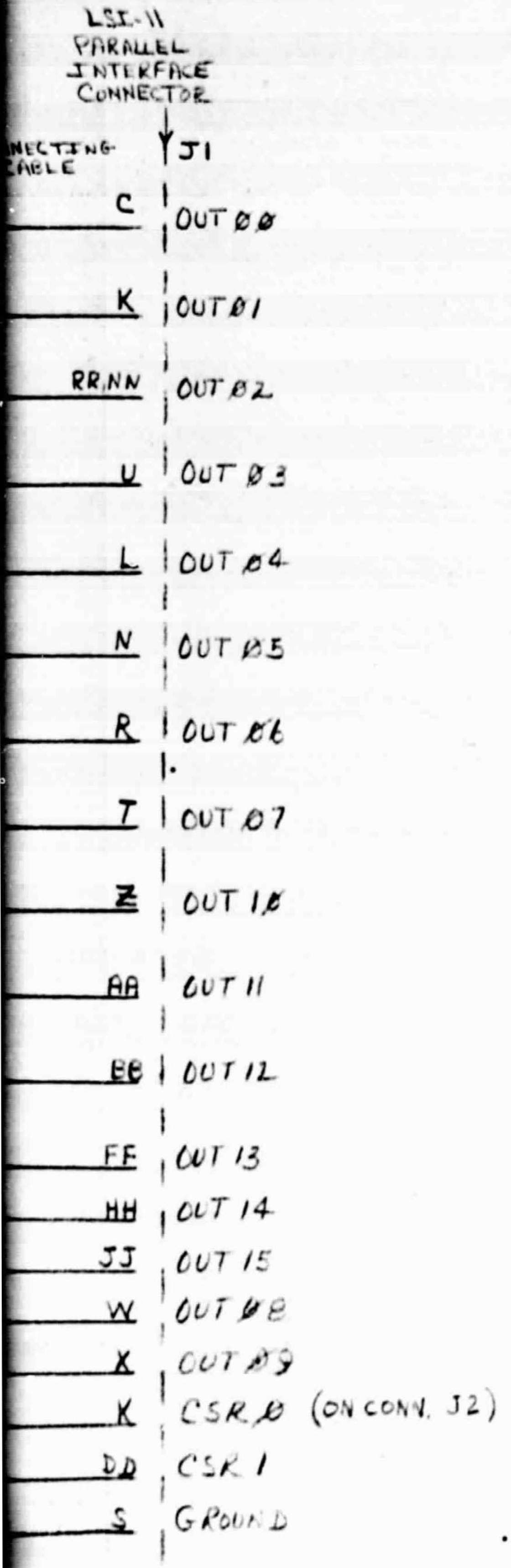
TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION
HOUSTON, TEXAS 77058

DRAWN BY M.H.UTLEY	DATE 10-26-79	TITLE CVDAS- CONTROL PANEL I/O INTERCONNECT PARALLEL INTERFACE INPUTS
DESIGN. ENG.		
PROJ. ENG.		DWG. NO. TH7942-1E03
		SHEET 1 OF 1



	CIRCUIT BOARD CONNECTOR		CHASSIS CONNECTOR	INTERCONNECTING RIBBON CABLE	LSI-11 PARALLEL INTERFAC CONNECTO	
	CI2	PI2	J1			J
CAL LIGHT	11	11	45	45	C	OU
REST LIGHT	12	12	43	43	K	OU
EX. LIGHT	13	13	27	27	RR,NN	OU
REC. LIGHT	14	14	38	38	U	O
	CI3	PI3				
PAUSE LIGHT	8	8	42	42	L	O
END LIGHT	9	9	41	41	N	OU
FVC LIGHT	10	10	40	40	R	OU
SPARE	11	11	39	39	T	OU
	CI6	PI6				
I DOWN	8	8	35	35	Z	OU
IUP	7	7	34	34	AA	OU
I FAST	6	6	33	33	EE	OU
ISLOW	9	9	31	31	FF	OU
		FOR FUTURE USE	30	30	HH	OU
			29	29	JJ	OU
CHASSIS CONNECTOR J3 (TO ANALOG INTERFACE/ CONTROL UNIT)		2 TEST TYPE Ø	37	37	W	OU
		5 TEST TYPE 1	36	36	X	OU
		4 CSR Ø	14	14	K	CS
		1 CSR 1	28	28	DD	CS
		GROUND	32	32	S	GR

FOLDOUT FRAME

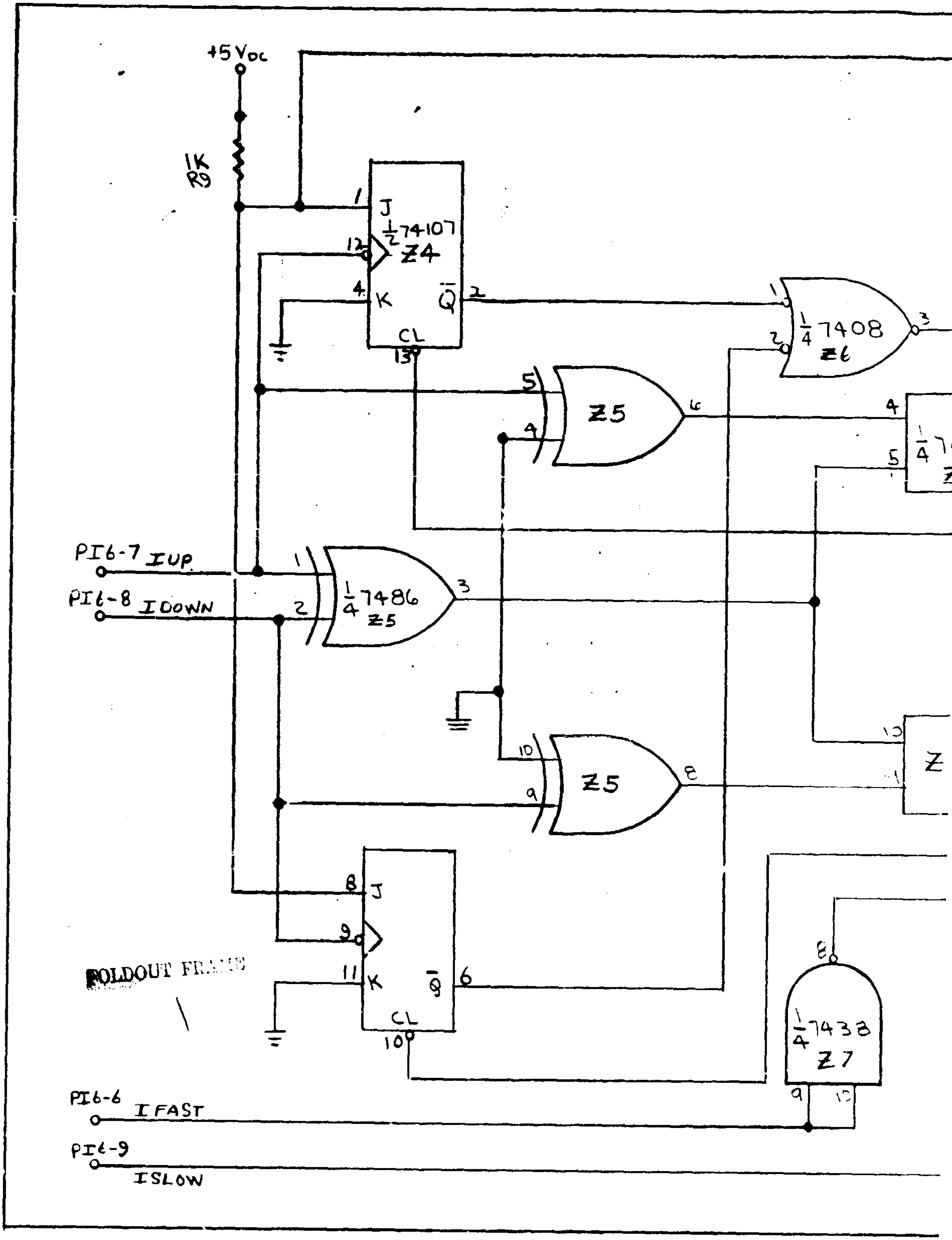


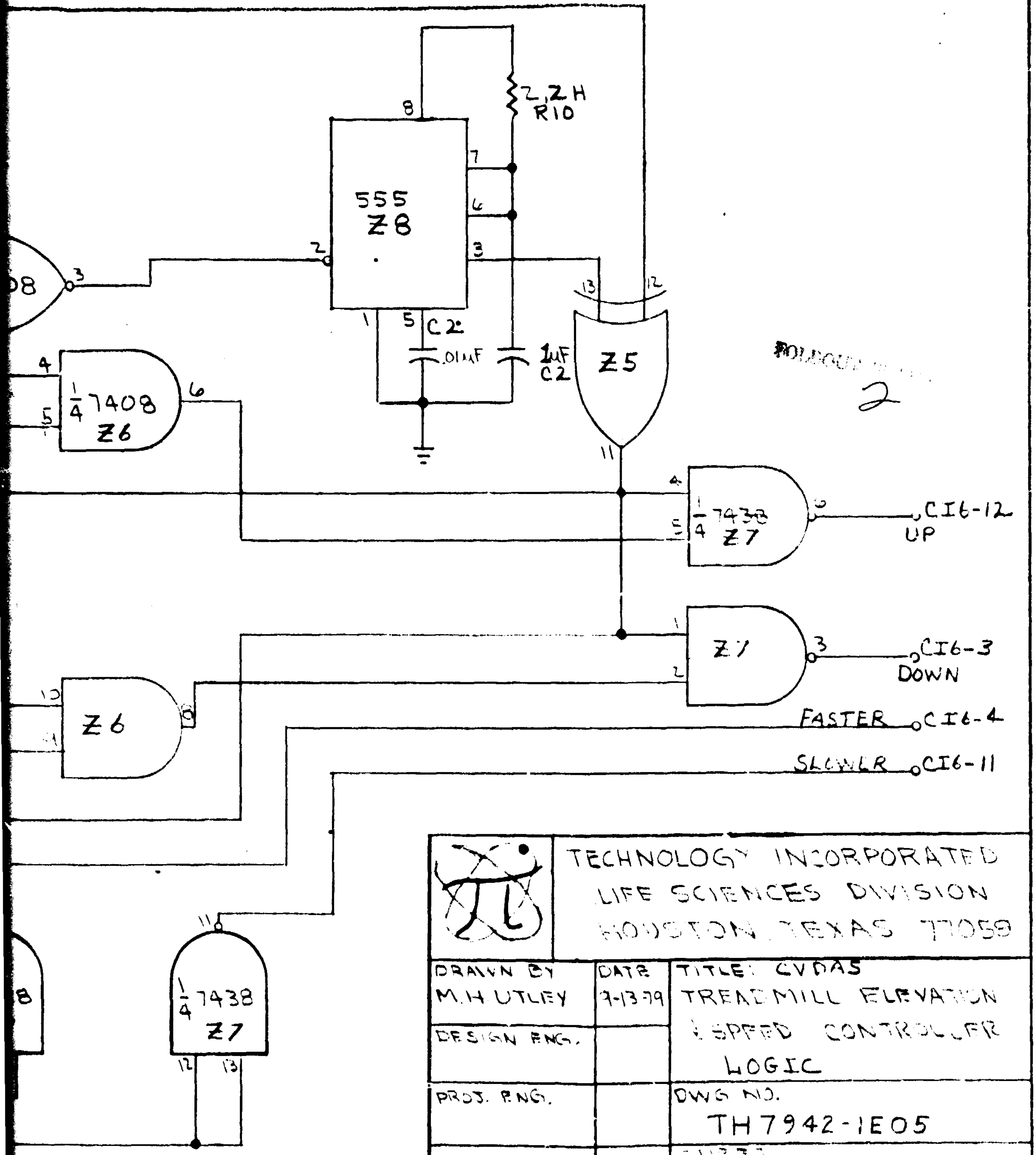
FOLDOUT FRAME

2

TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION
HOUSTON, TEXAS 77058

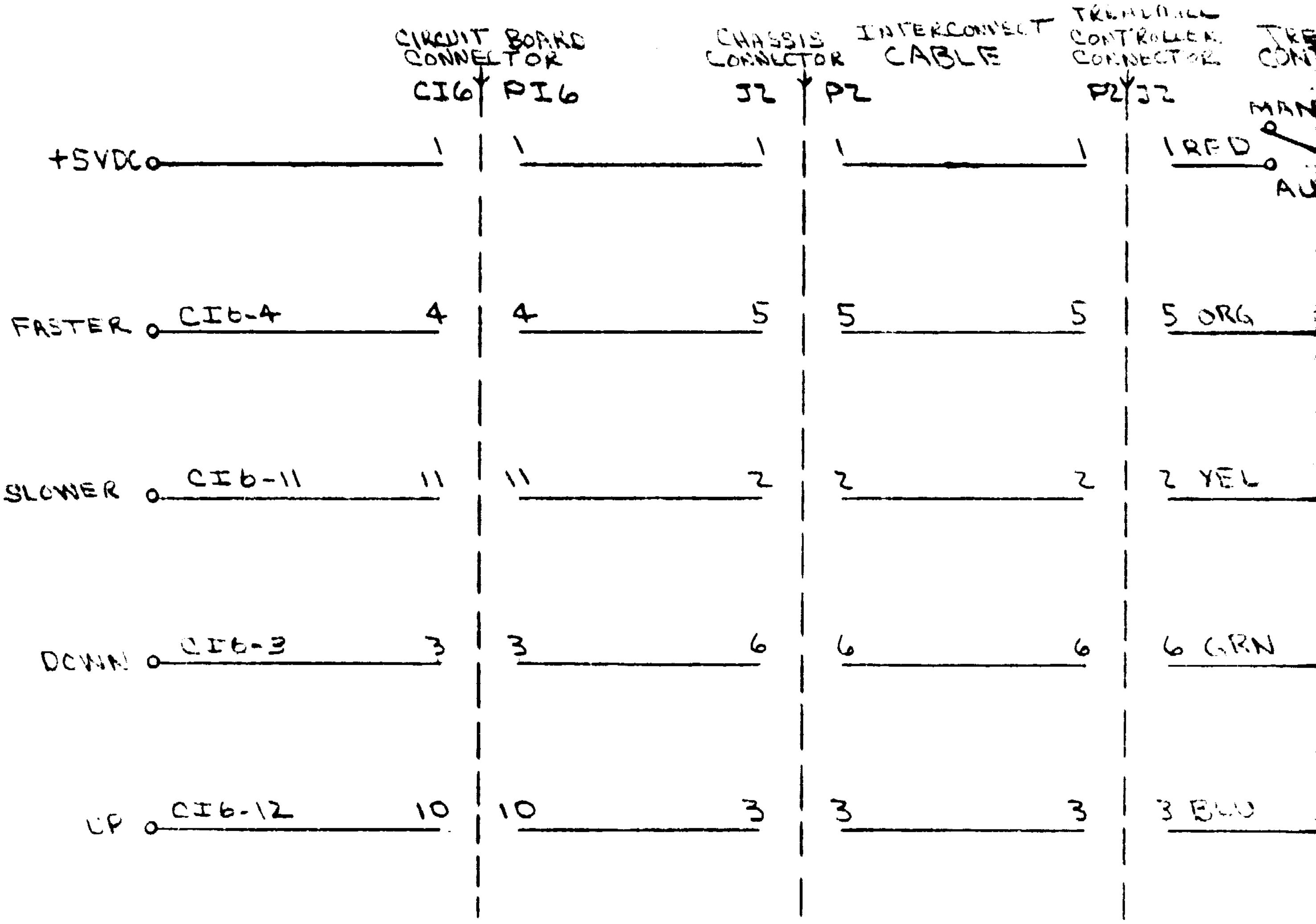
DRAWN BY M.H. UTLEY	DATE 10 25 79	TITLE CVDAS- CONTROL PANEL I/O INTERCONNECT PARALLEL INTERFACE OUTPUTS
DESIGN ENG.		
PROJ. ENG.		DWG. NO. TH7942-1E04
		SHEET 1 OF 1


+5 Vdc
1K
R9
1
J
½ 74107
Z4
12
4
K
Q̄
2
CL
13
¼ 7408
Z6
1
2
3
5
4
Z5
6
4
5
¼ 7
PI6-7 IUP
PI6-8 IDOWN
1
2
¼ 7486
Z5
3
10
9
Z5
8
Z
8
J
9
11
K
Q̄
6
CL
10
8
¼ 7438
Z7
9
10
PI6-6 IFAST
PI6-9 ISLOW

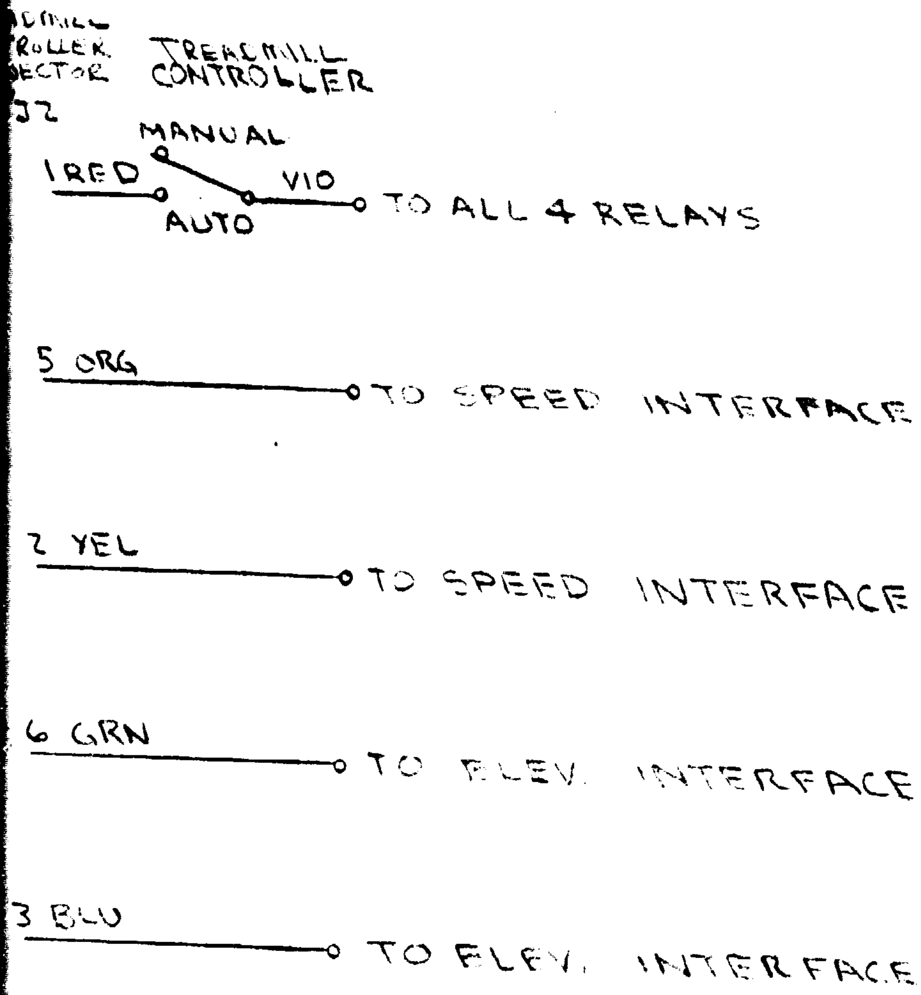


FOLDOUT FRAME 2

TECHNOLOGY INCORPORATED LIFE SCIENCES DIVISION HOUSTON TEXAS 77058		
DRAWN BY M.H. UTLEY	DATE 9-13-79	TITLE: CVDAS TREADMILL ELEVATION & SPEED CONTROLLER LOGIC
DESIGN ENG.		
PROJ. ENG.		DWG NO. TH7942-1E05
		SHEET 1 OF 1



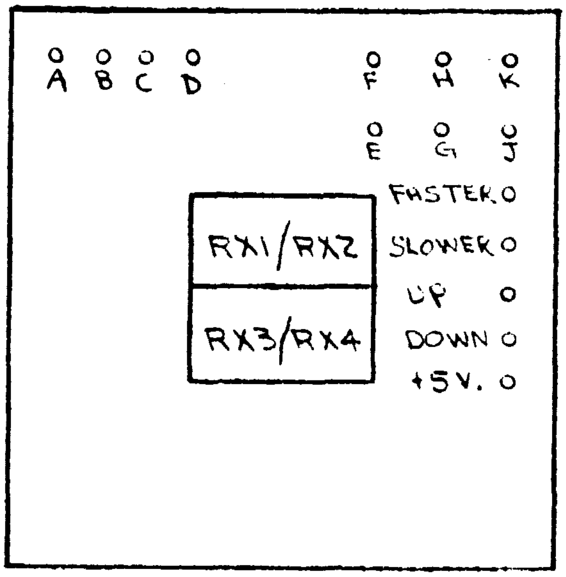
FOLDOUT FRAME

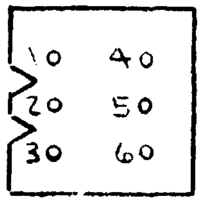


2

TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION
HOUSTON, TEXAS 77058

DRAWN BY M.H.UTLEY	DATE 10-26-79	TITLE CVDAS-CONTROL PANEL TO TREADMILL INTERCONNECT
DESIGN ENG.		
PROJ. ENG.		DWG NO TH7942-1E06
		SHEET 1 OF 1





FOLDOUT FRAME

A	YEL
B	ORG.
C	RED
D	BRW.
E	BLU
F	GRN
G	GRAY
H	VIO
J	BLK
K	WHT

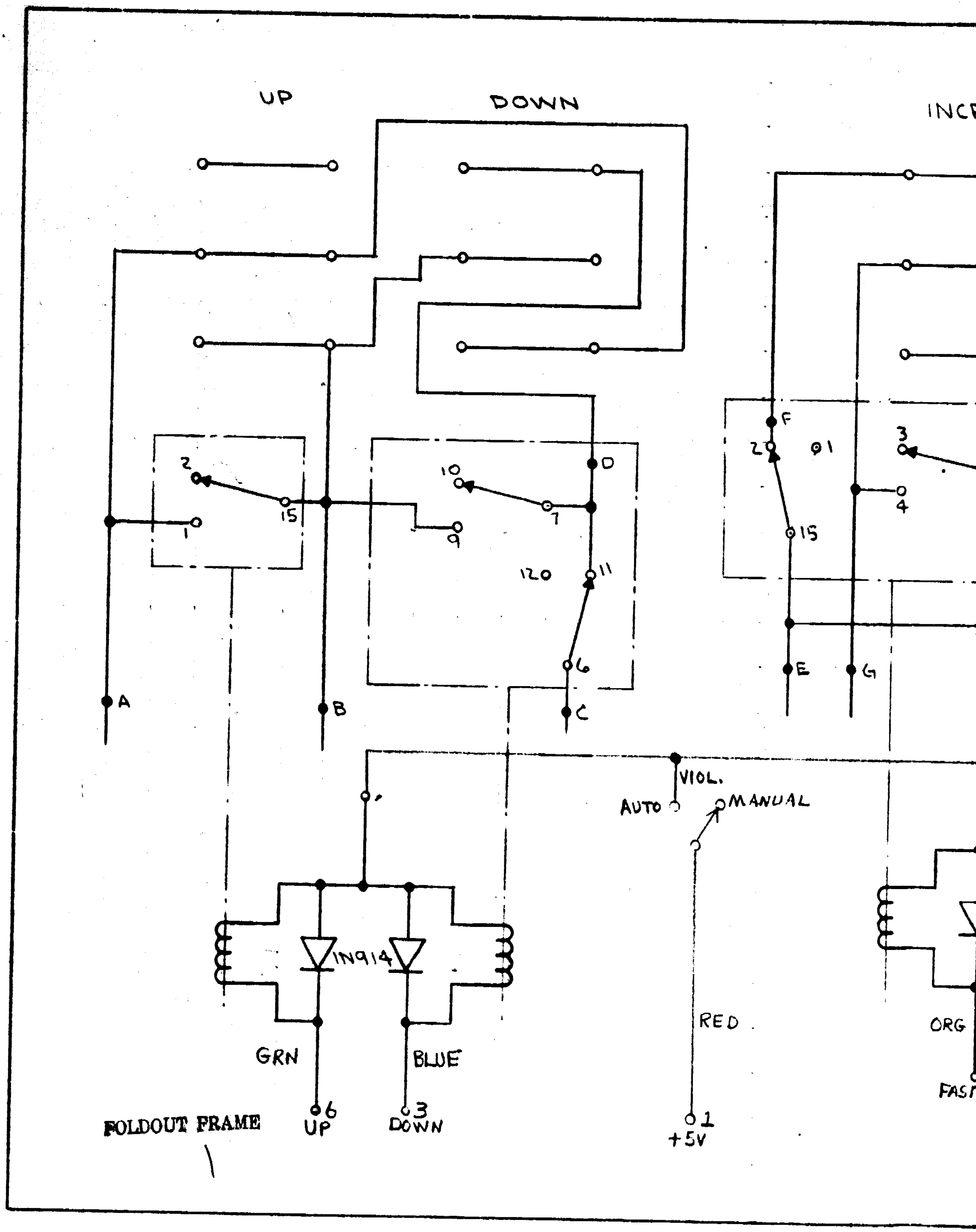
FASTER	ORG	- PIN 5
SLOWER	YEL	- PIN 2
UP	GRN	- PIN 6
DOWN	BLU	- PIN 3
+5 VOL	VIO	- PIN 1
SPARE	BRN	- PIN 4

FOLDOUT FRAME

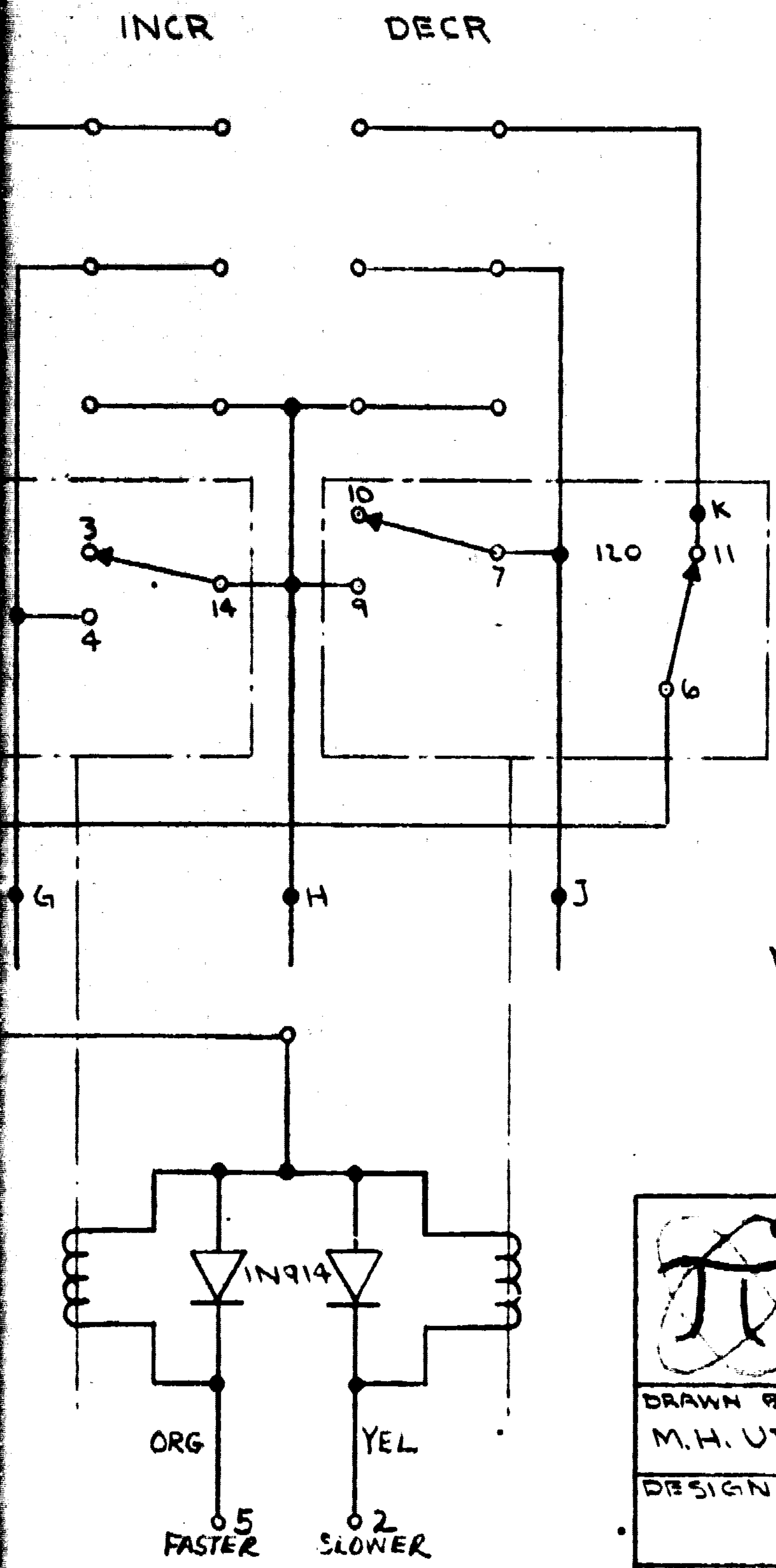
2

TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION
HOUSTON, TEXAS 77058

DRAWN BY M.H UTLEY	DATE 10-24-79	TITLE CVDAS- LAYOUT-TREADMILL CONTROLLER MOD. BOARD
DESIGN ENG.		
PROJ. ENG.		DWG. NO TH7942-1A03
		SHEET 1 OF 1



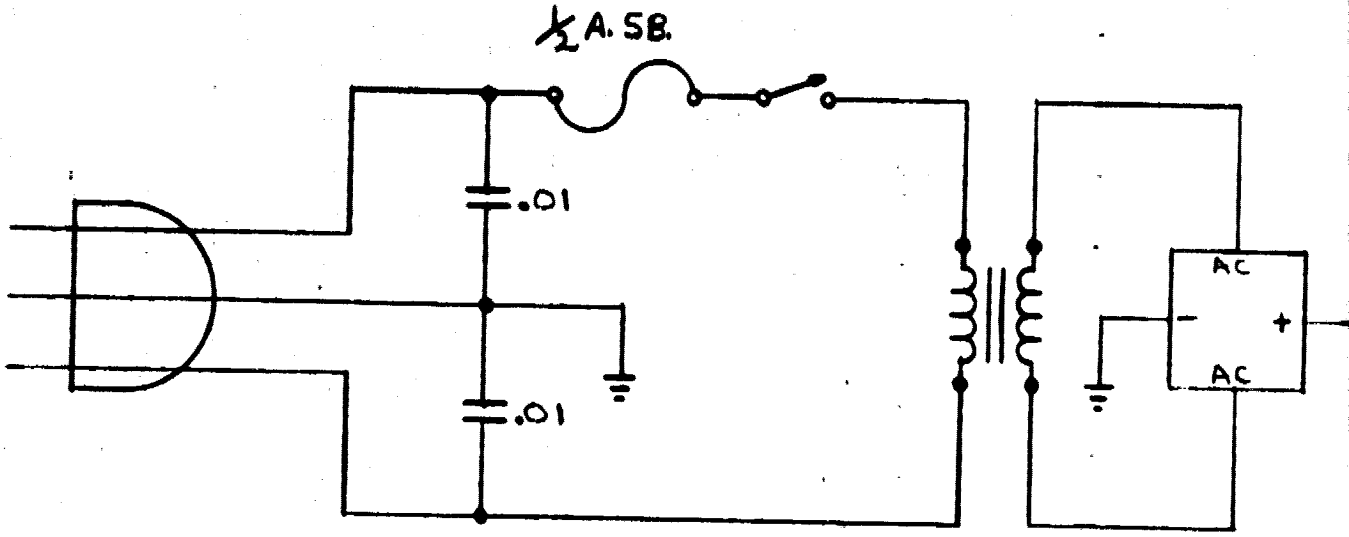
FOLDOUT FRAME

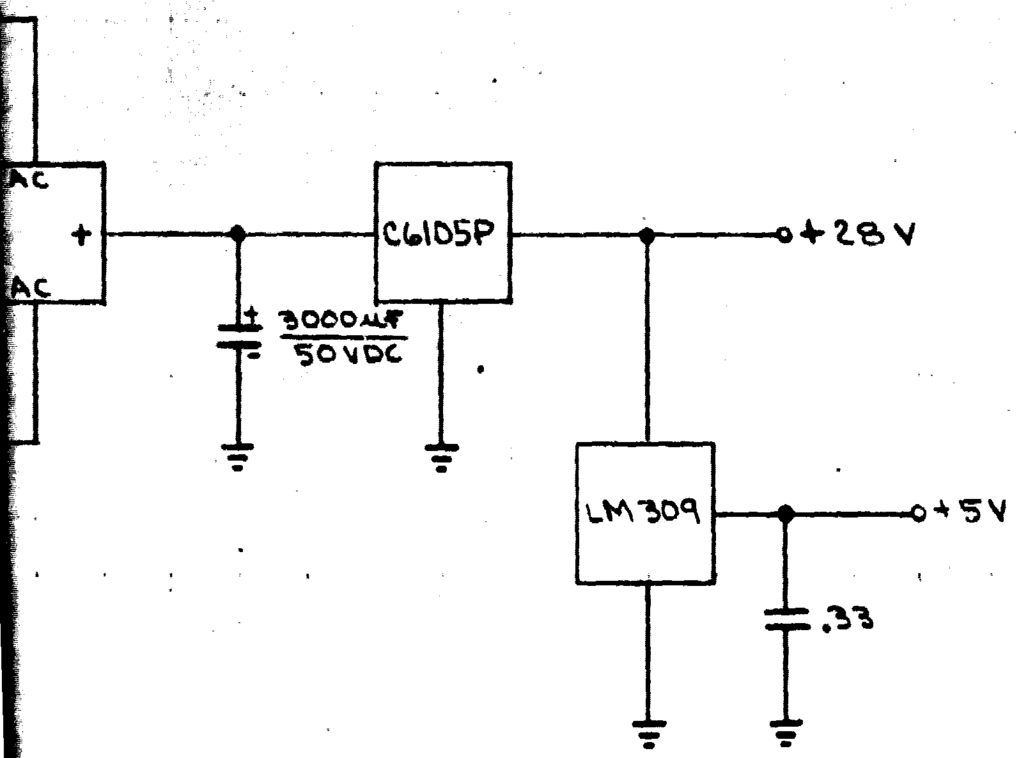


NOTE:(1) +5VDC IS CONNECTED THRU
MANUAL/ AUTO SWITCH.

TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION
HOUSTON, TEXAS 77058

DRAWN BY M.H. UTLEY	DATE 10-24-79	TITLE CVDAS-TREADMILL CONTROLLER MODIFICATION
DESIGN ENG.		
PROJ. ENG.		DWG. NO. TH7942-1E07
		SHEET 1 OF 1


½ A. SB.
.01
.01
AC
-
+
AC



FOLDOUT

2

TECHNOLOGY INCORPORATED
LIFE SCIENCES DIVISION
HOUSTON, TEXAS 77058

DRAWN BY M.H.UTLEY	DATE 9-3079	TITLE CVDAS - CONTROL PANEL POWER SUPPLY
DESIGN ENG.		
PROJ. ENG		DWG. NO. TH7942-1E08
		SHEET 1 OF 1

APPENDIX G

FLOWCHARTS OF THE SOFTWARE

Hierarchy Diagram of the CDAS PROGRAMS

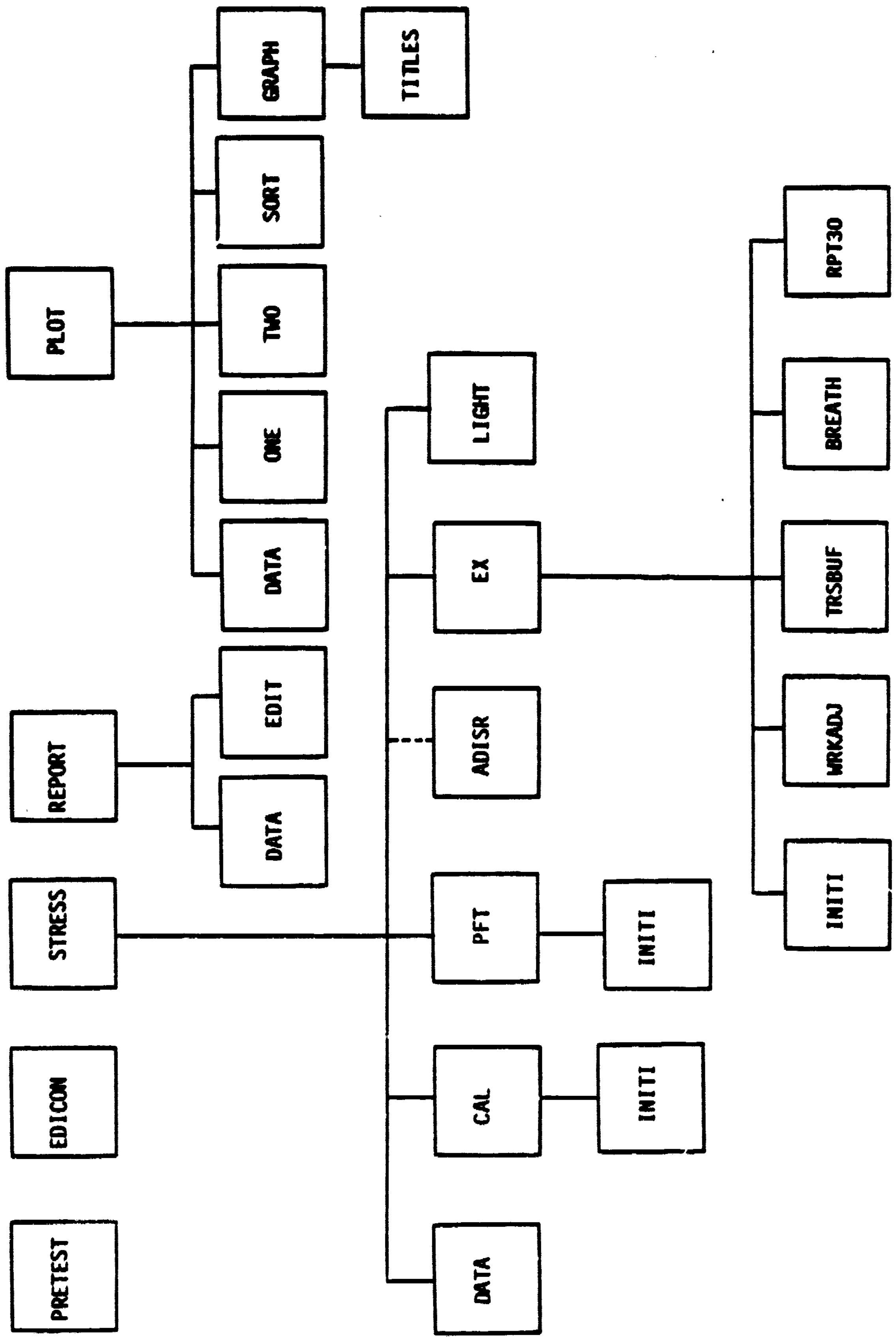
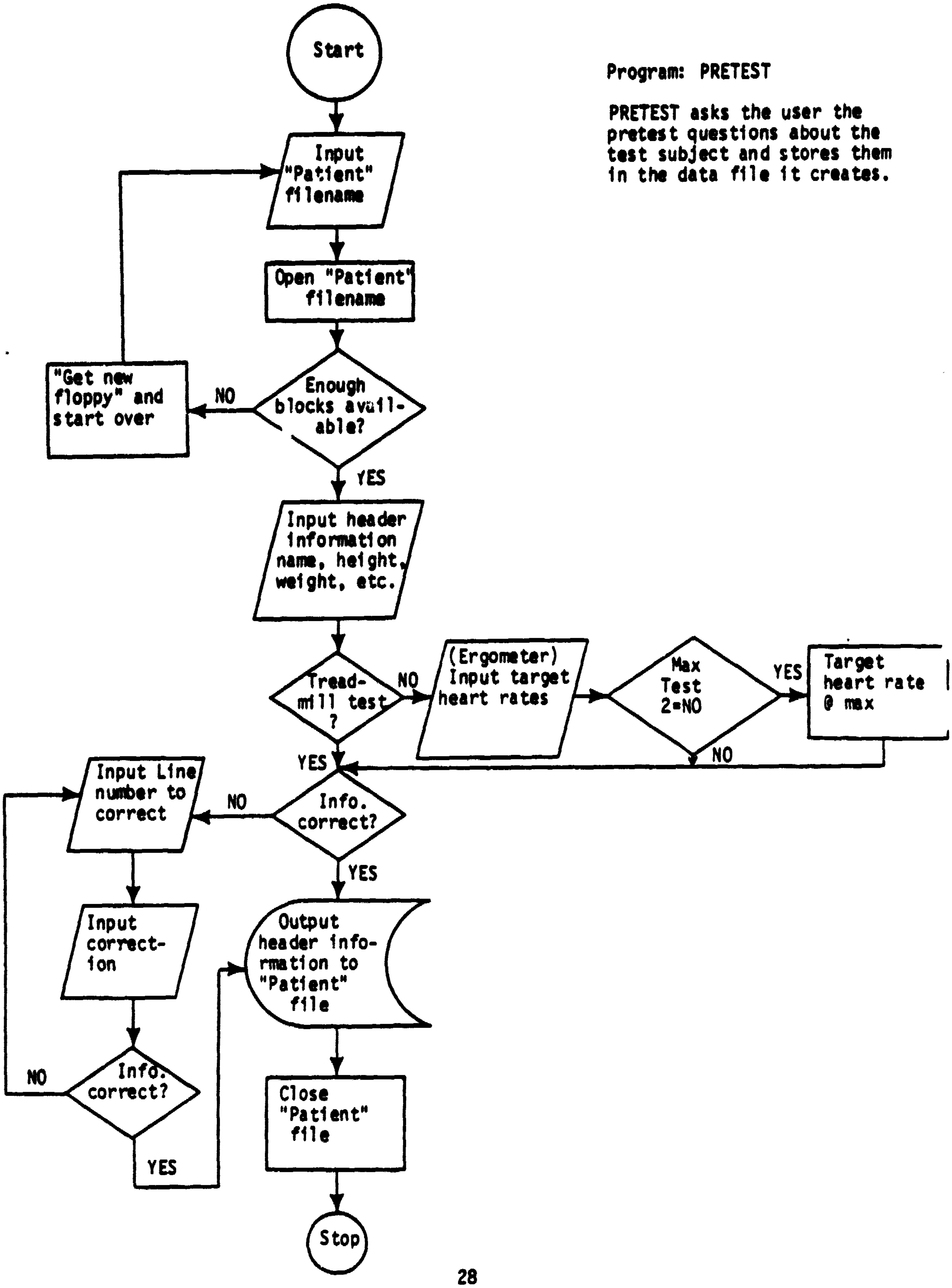
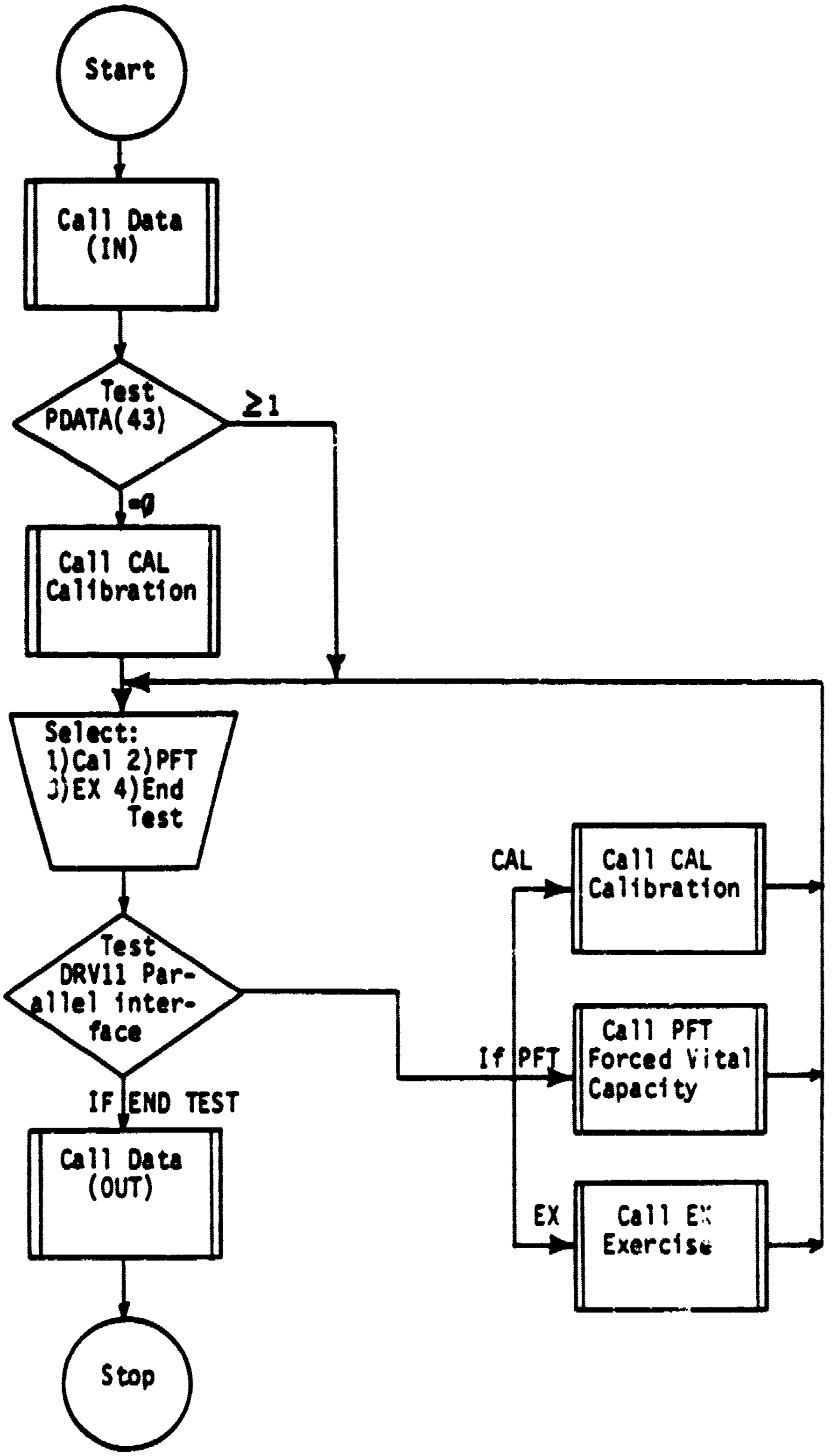


Figure 6


Start
Input "Patient" filename
Open "Patient" filename
Enough blocks avail-able?
NO
"Get new floppy" and start over
YES
Input header information name, height, weight, etc.
Tread-mill test ?
NO
(Ergometer) Input target heart rates
Max Test 2=NO
YES
Target heart rate @ max
NO
YES
Info. correct?
NO
Input Line number to correct
Input correct-ion
Info. correct?
NO
YES
YES
Output header info-rmation to "Patient" file
Close "Patient" file
Stop

Program: PRETEST

PRETEST asks the user the pretest questions about the test subject and stores them in the data file it creates.

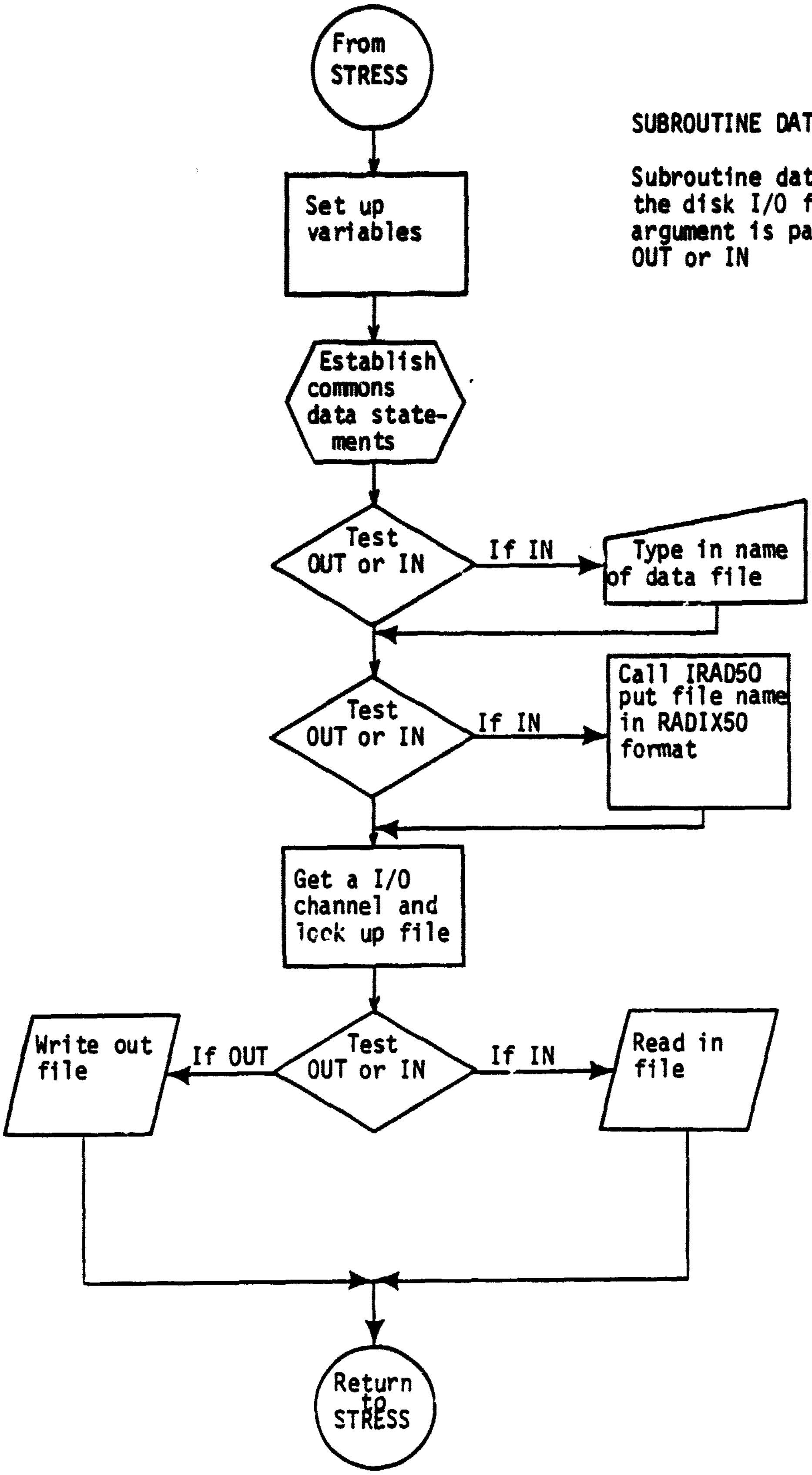


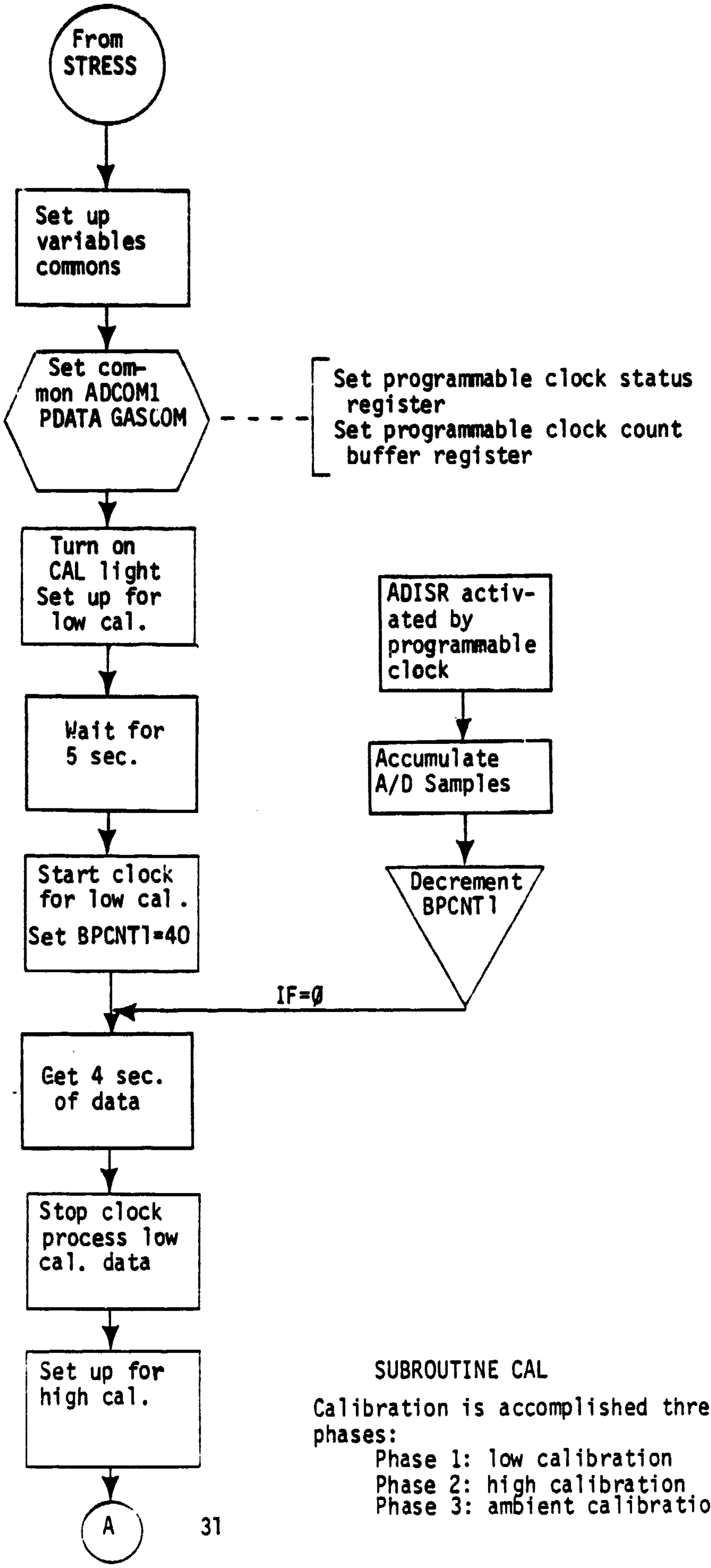
Flow Chart of STRESS

STRESS is the main program that tests the CDAS front pannel switches; only those lite are functional.

SUBROUTINE DATA

Subroutine data performs all the disk I/O for CDAS one argument is passed, either OUT or IN

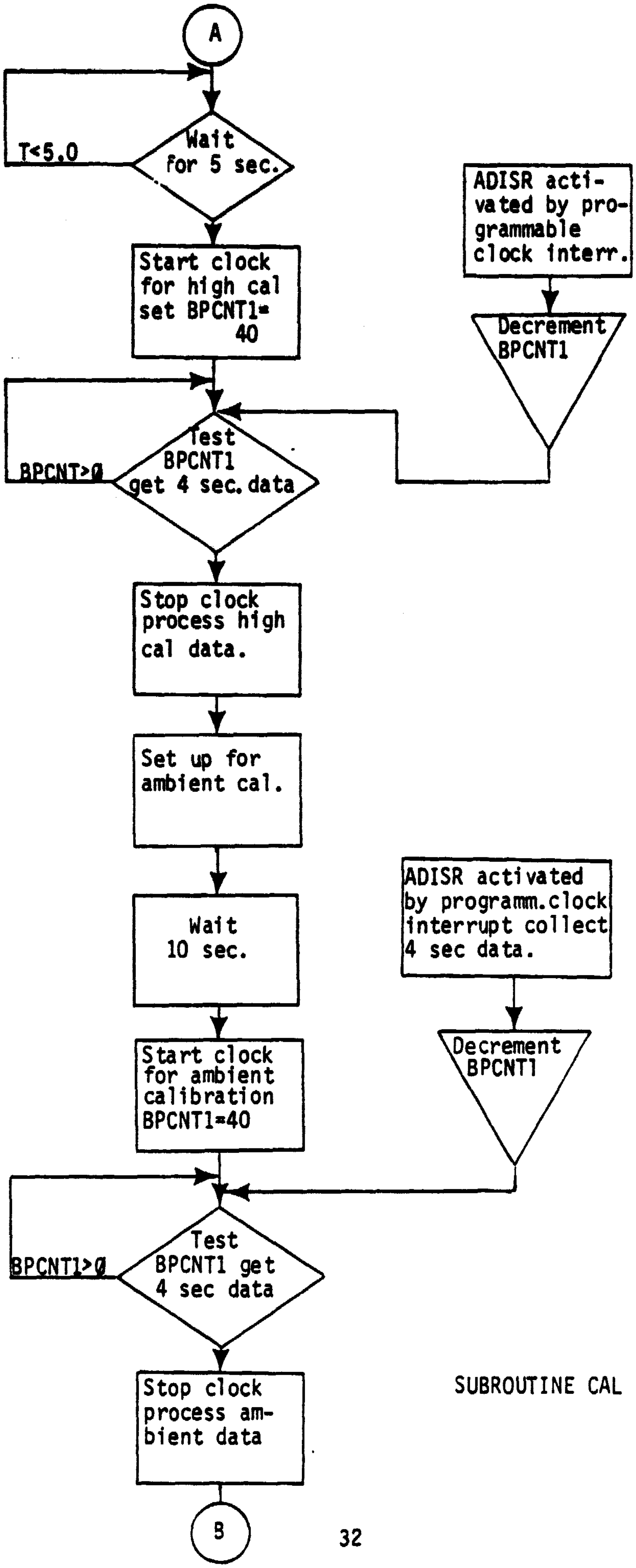

From STRESS
Set up variables
Establish commons data statements
Test OUT or IN
If IN
Type in name of data file
Test OUT or IN
If IN
Call IRAD50 put file name in RADIX50 format
Get a I/O channel and lcck up file
Test OUT or IN
If OUT
Write out file
If IN
Read in file
Return to STRESS


From STRESS
Set up variables commons
Set common ADCOM1 PDATA GASCOM
Set programmable clock status register
Set programmable clock count buffer register
Turn on CAL light Set up for low cal.
Wait for 5 sec.
Start clock for low cal.
Set BPCNT1=40
ADISR activated by programmable clock
Accumulate A/D Samples
Decrement BPCNT1
IF=Ø
Get 4 sec. of data
Stop clock process low cal. data
Set up for high cal.
A

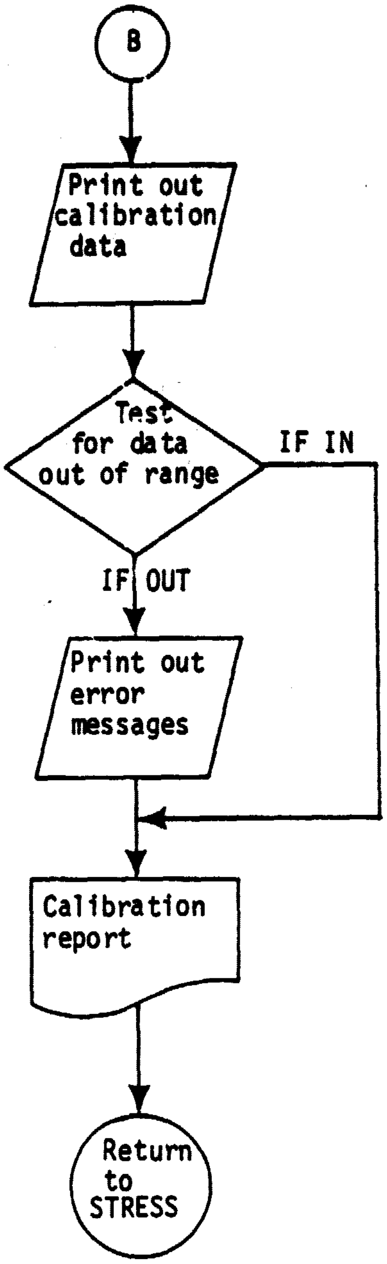
SUBROUTINE CAL

Calibration is accomplished three phases:

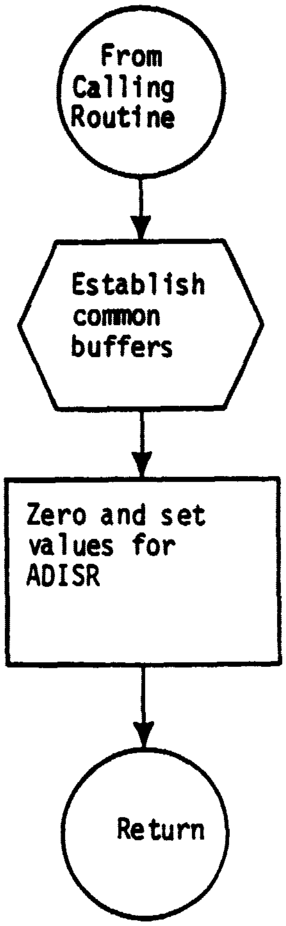
Phase 1: low calibration
Phase 2: high calibration
Phase 3: ambient calibration


A
Wait for 5 sec.
T<5.0
Start clock for high cal set BPCNT1= 40
ADISR activated by programmable clock interr.
Decrement BPCNT1
Test BPCNT1 get 4 sec. data
BPCNT>Ø
Stop clock process high cal data.
Set up for ambient cal.
Wait 10 sec.
ADISR activated by programm.clock interrupt collect 4 sec data.
Start clock for ambient calibration BPCNT1=40
Decrement BPCNT1
Test BPCNT1 get 4 sec data
BPCNT1>Ø
Stop clock process ambient data
B

SUBROUTINE CAL (Continued)

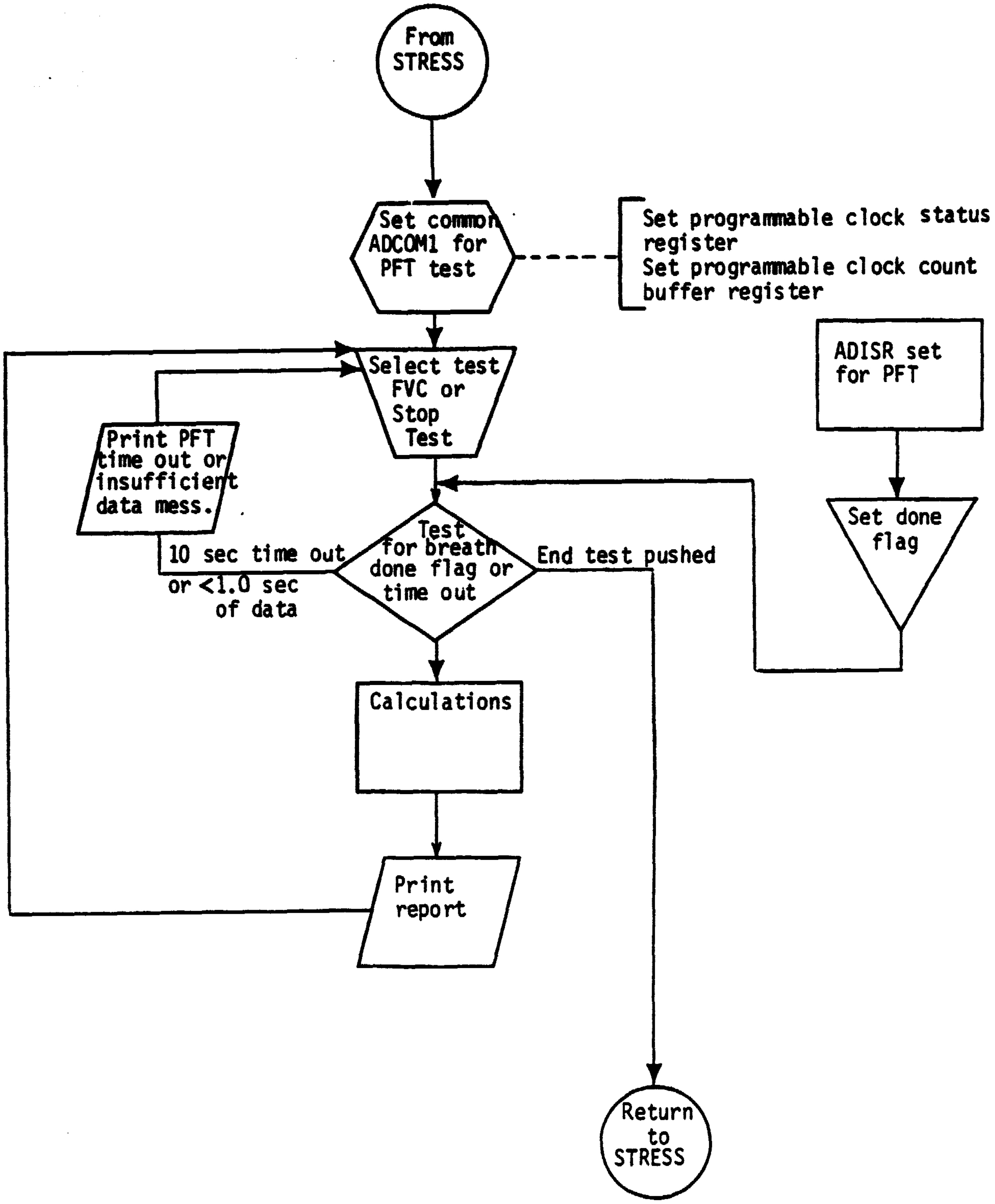


SUBROUTINE CAL CONTINUED



SUBROUTINE INITI

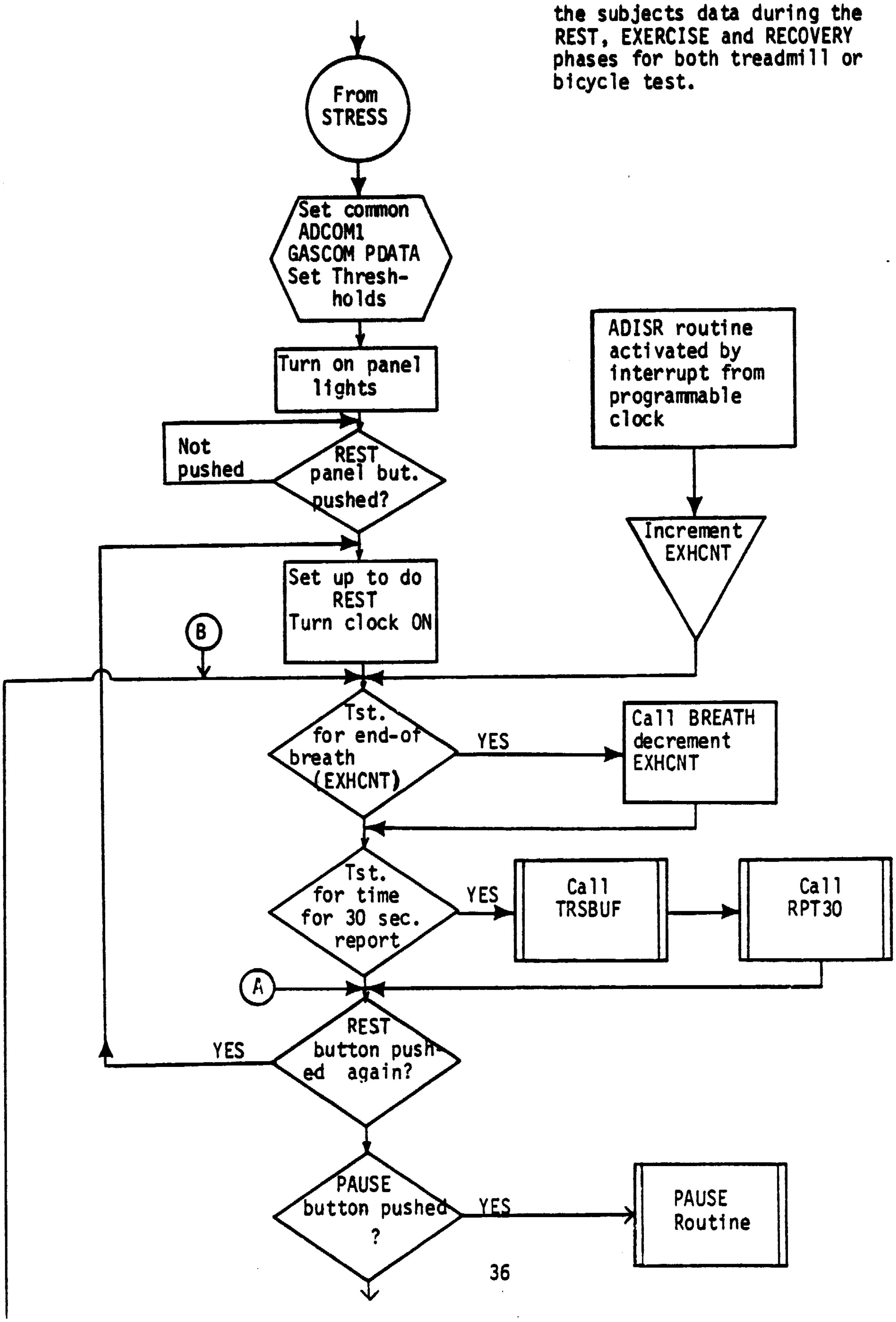
This subroutine is used by CAL. PFT and EX subroutines to restablish the value in the ADCOM1 common area used by ADISR.

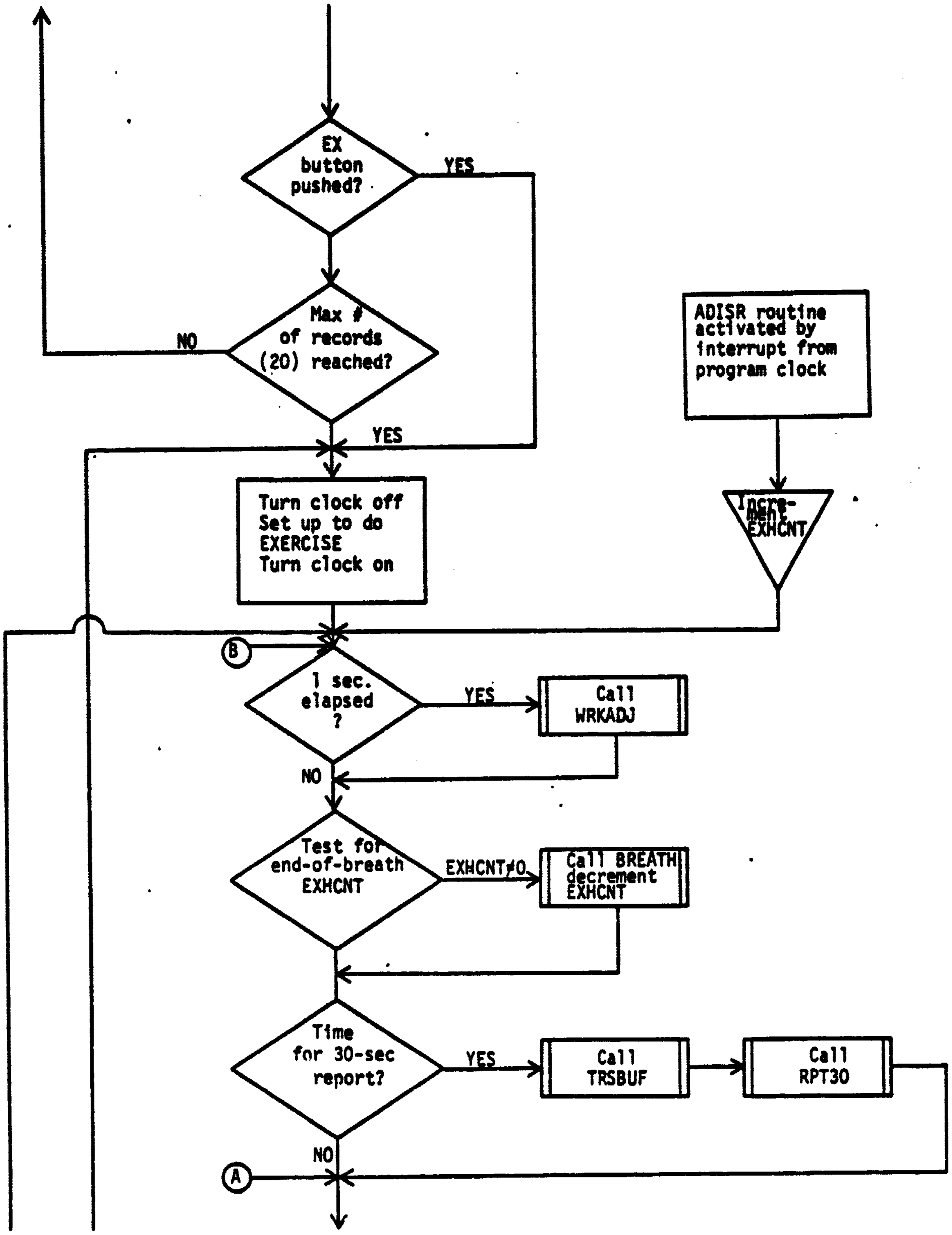

From STRESS
Set common ADCOM1 for PFT test
Set programmable clock status register
Set programmable clock count buffer register
Select test FVC or Stop Test
ADISR set for PFT
Print PFT time out or insufficient data mess.
Set done flag
10 sec time out or <1.0 sec of data
Test for breath done flag or time out
End test pushed
Calculations
Print report
Return to STRESS

Flow Chart for SUBROUTINE PFT

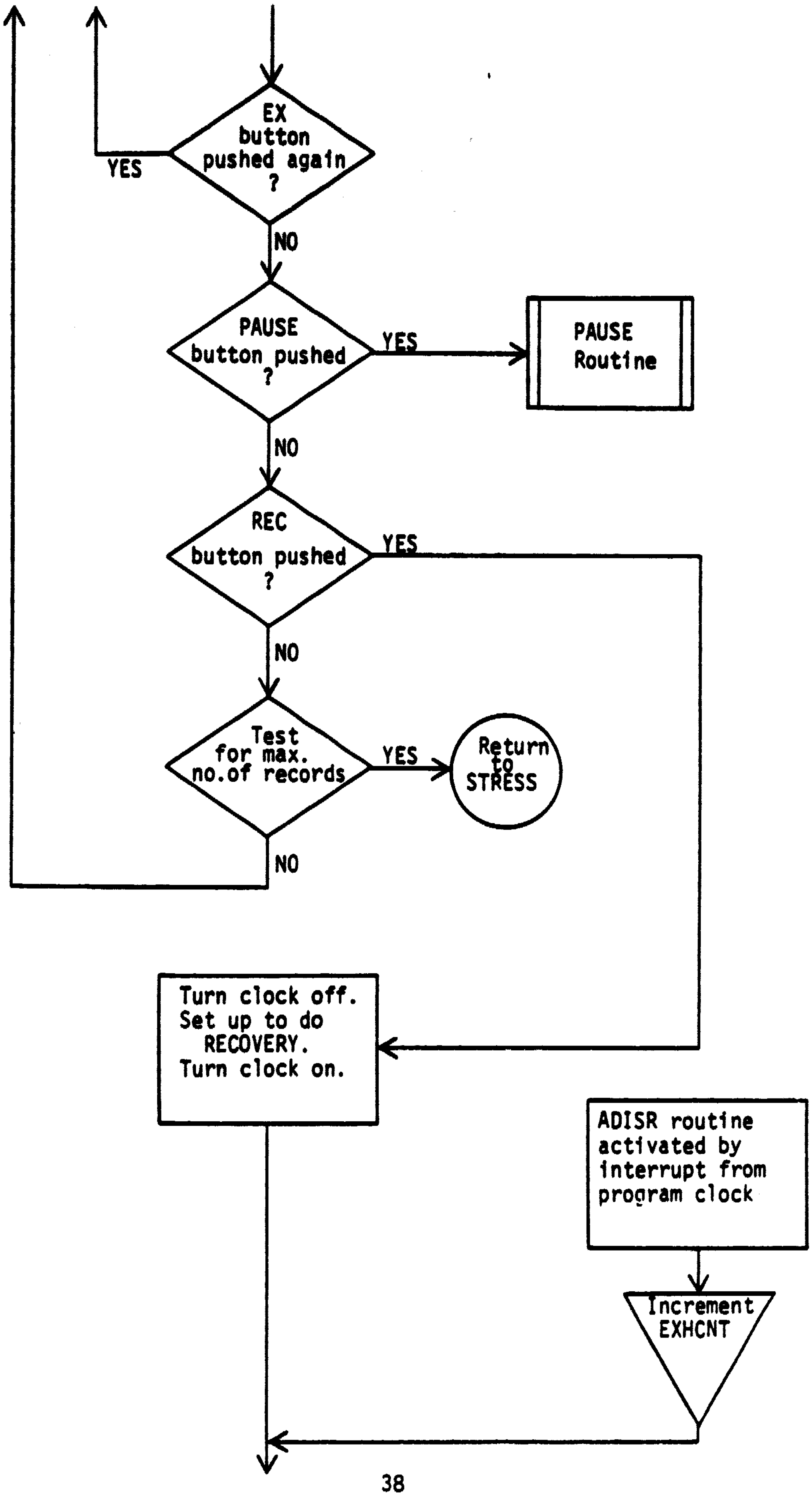
SUBROUTINE EX

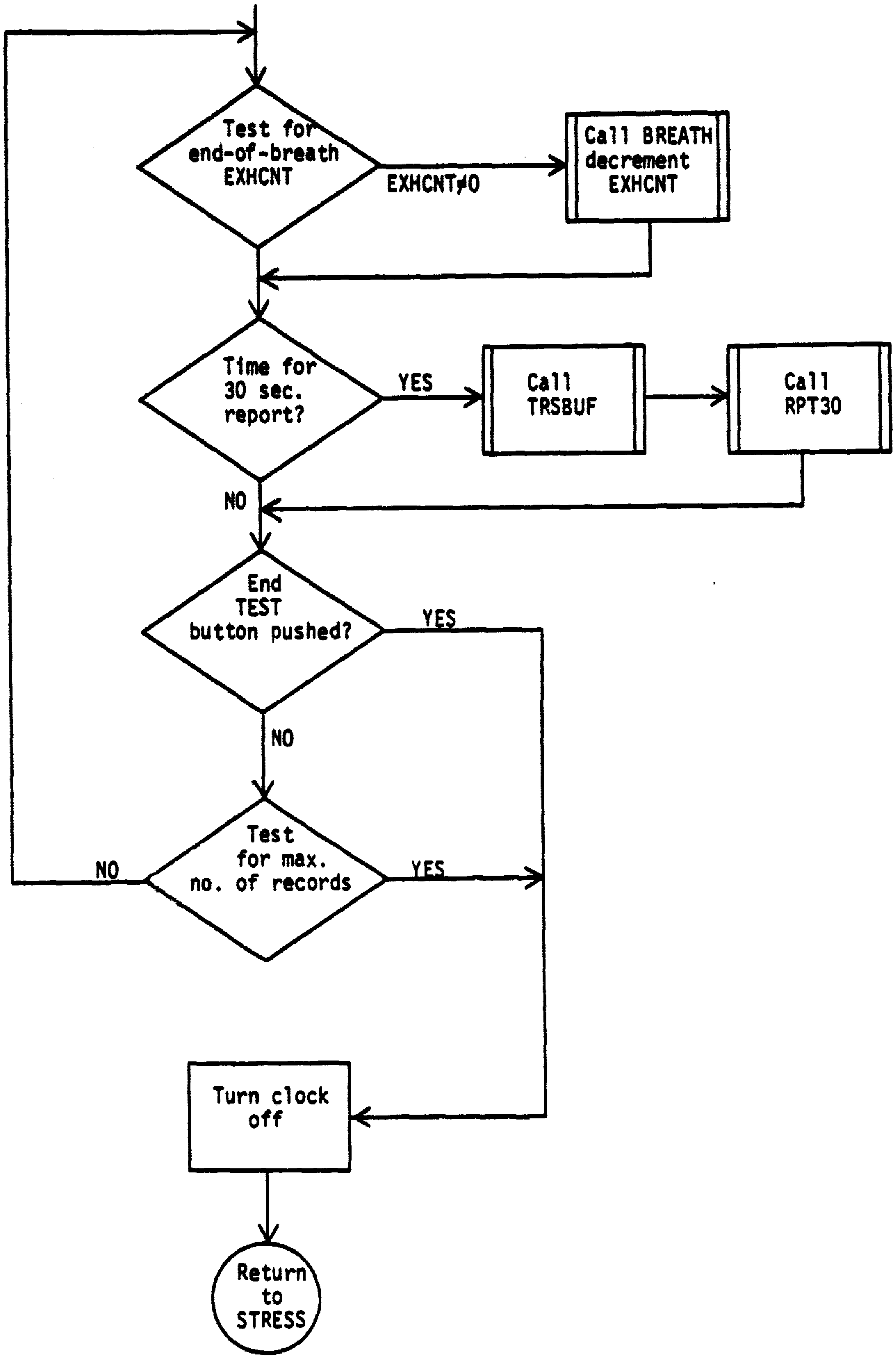
The exercise subroutine collects the subjects data during the REST, EXERCISE and RECOVERY phases for both treadmill or bicycle test.


From STRESS
Set common ADCOM1 GASCOM PDATA Set Thresh-holds
Turn on panel lights
Not pushed
REST panel but. pushed?
ADISR routine activated by interrupt from programmable clock
Increment EXHCNT
Set up to do REST Turn clock ON
B
Tst. for end-of breath (EXHCNT)
YES
Call BREATH decrement EXHCNT
Tst. for time for 30 sec. report
YES
Call TRSBUF
Call RPT30
A
REST button push-ed again?
YES
PAUSE button pushed ?
YES
PAUSE Routine



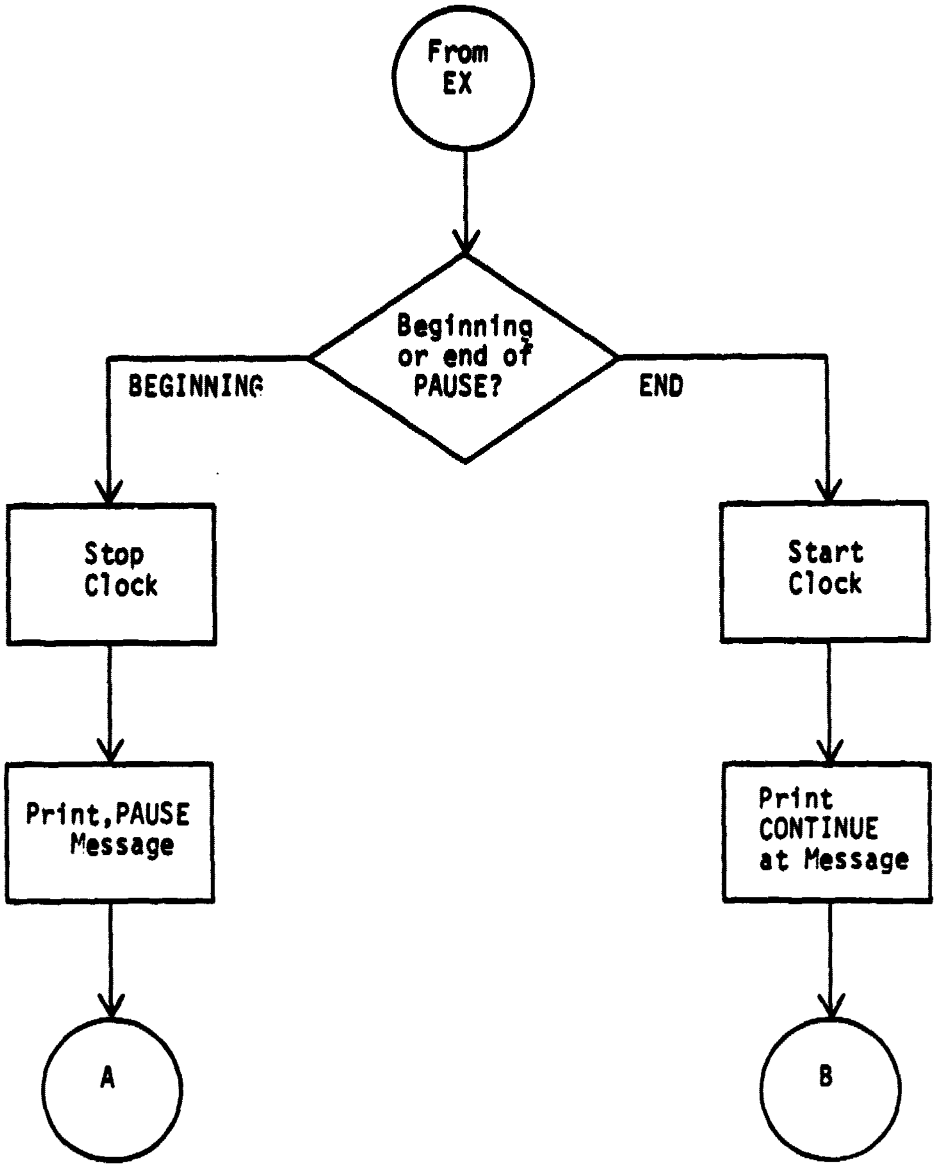
EX button pushed?
YES
Max # of records (20) reached?
NO
YES
ADISR routine activated by interrupt from program clock
Turn clock off
Set up to do EXERCISE
Turn clock on
Incre-ment EXHCNT
B
1 sec. elapsed ?
YES
Call WRKADJ
NO
Test for end-of-breath EXHCNT
EXHCNT≠0
Call BREATH decrement EXHCNT
Time for 30-sec report?
YES
Call TRSBUF
Call RPT30
NO
A


EX button pushed again ?
YES
NO
PAUSE button pushed ?
YES
PAUSE Routine
NO
REC button pushed ?
YES
NO
Test for max. no.of records
YES
Return to STRESS
NO
Turn clock off. Set up to do RECOVERY. Turn clock on.
ADISR routine activated by interrupt from program clock
Increment EXHCNT


Test for end-of-breath EXHCNT
EXHCNT≠0
Call BREATH decrement EXHCNT
Time for 30 sec. report?
YES
Call TRSBUF
Call RPT30
NO
End TEST button pushed?
YES
NO
Test for max. no. of records
NO
YES
Turn clock off
Return to STRESS

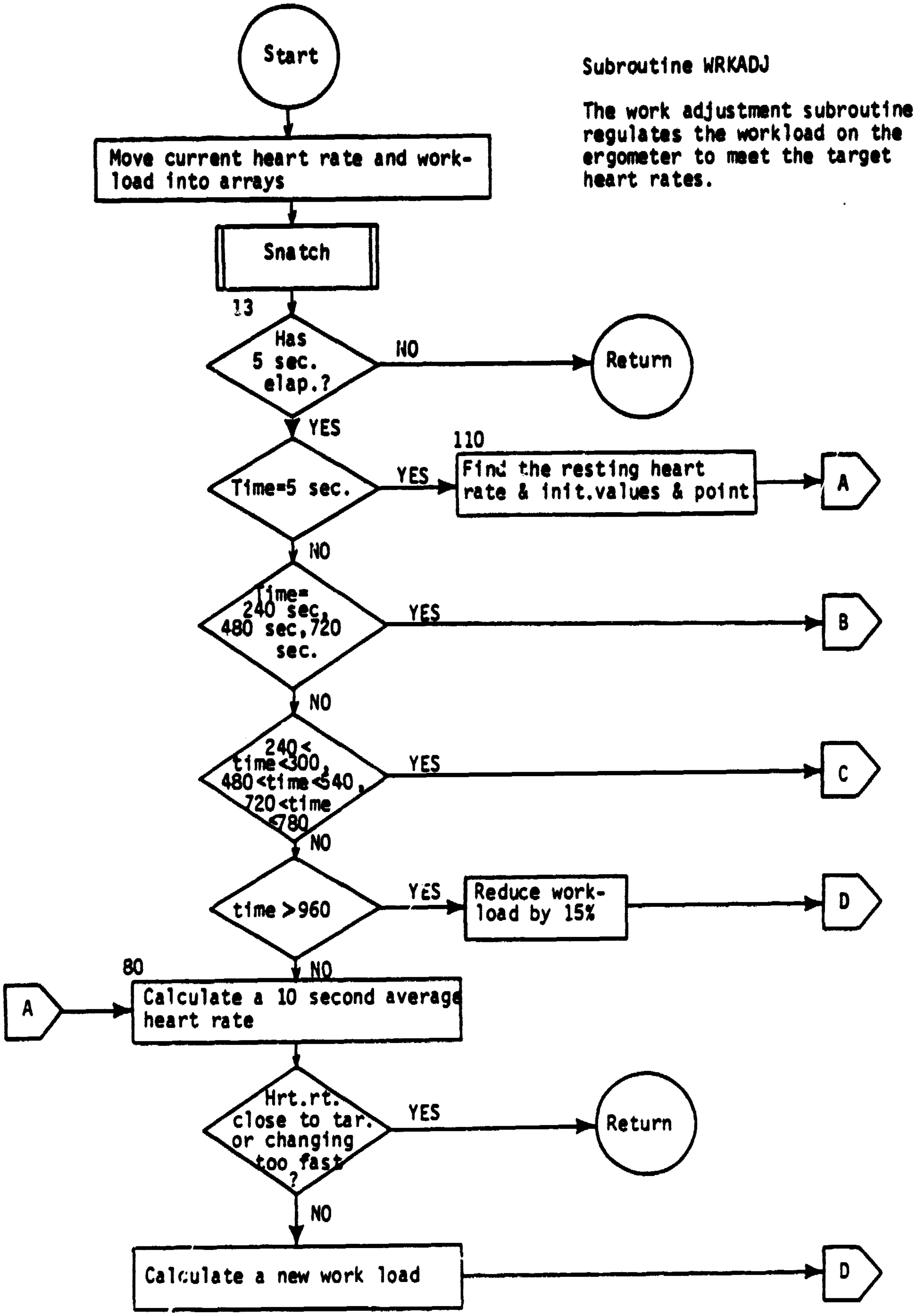
Routine PAUSE
for SUBROUTINE EX

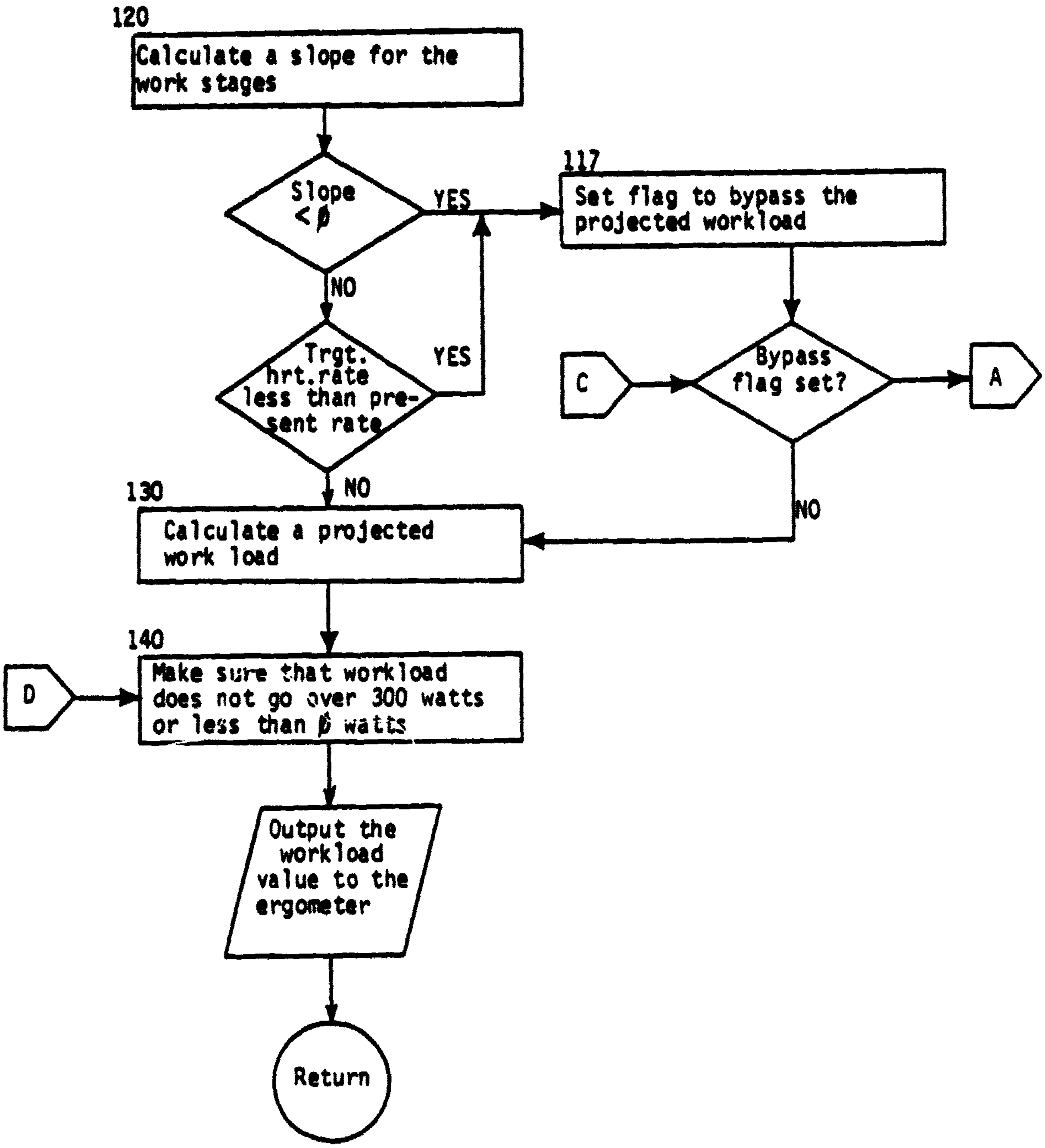
This routine is called when the pause button is pushed.



Subroutine WRKADJ

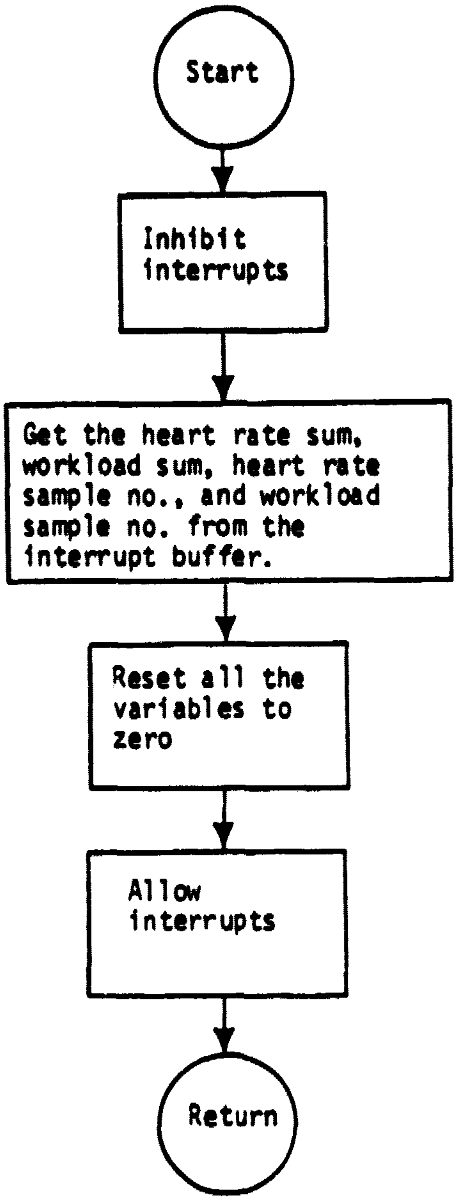
The work adjustment subroutine regulates the workload on the ergometer to meet the target heart rates.

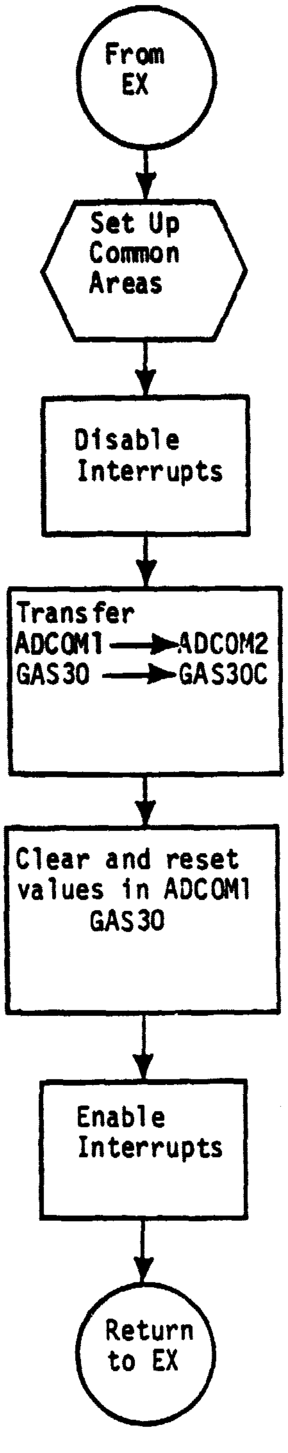

Start
Move current heart rate and work-load into arrays
Snatch
13
Has 5 sec. elap.?
NO
Return
YES
110
Time=5 sec.
YES
Find the resting heart rate & init.values & point
A
NO
Time= 240 sec, 480 sec,720 sec.
YES
B
NO
240< time<300, 480<time<540, 720<time <780
YES
C
NO
time >960
YES
Reduce work-load by 15%
D
NO
80
A
Calculate a 10 second average heart rate
Hrt.rt. close to tar. or changing too fast ?
YES
Return
NO
Calculate a new work load
D


120
Calculate a slope for the work stages
Slope < ∅
YES
NO
117
Set flag to bypass the projected workload
Trgt. hrt.rate less than present rate
YES
NO
C
Bypass flag set?
A
NO
130
Calculate a projected work load
140
D
Make sure that workload does not go over 300 watts or less than ∅ watts
Output the workload value to the ergometer
Return

Subroutine SNATCH

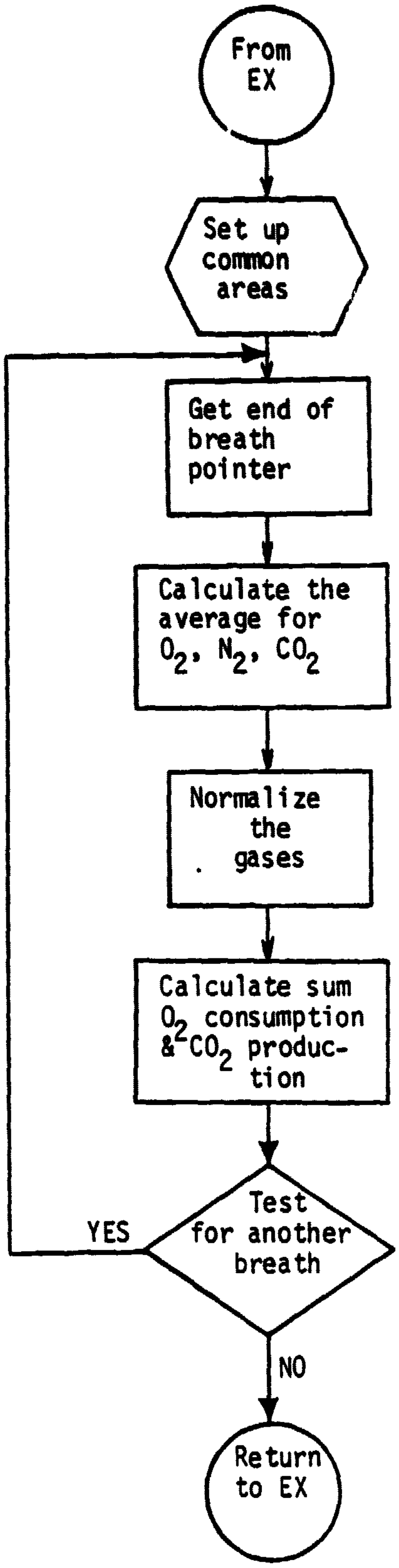
This routine prevents the A/D interrupt routine from updating the heart rate and workload data to maintain accuracy while it is being moved to the WRKADJ routine


Start
Inhibit interrupts
Get the heart rate sum, workload sum, heart rate sample no., and workload sample no. from the interrupt buffer.
Reset all the variables to zero
Allow interrupts
Return



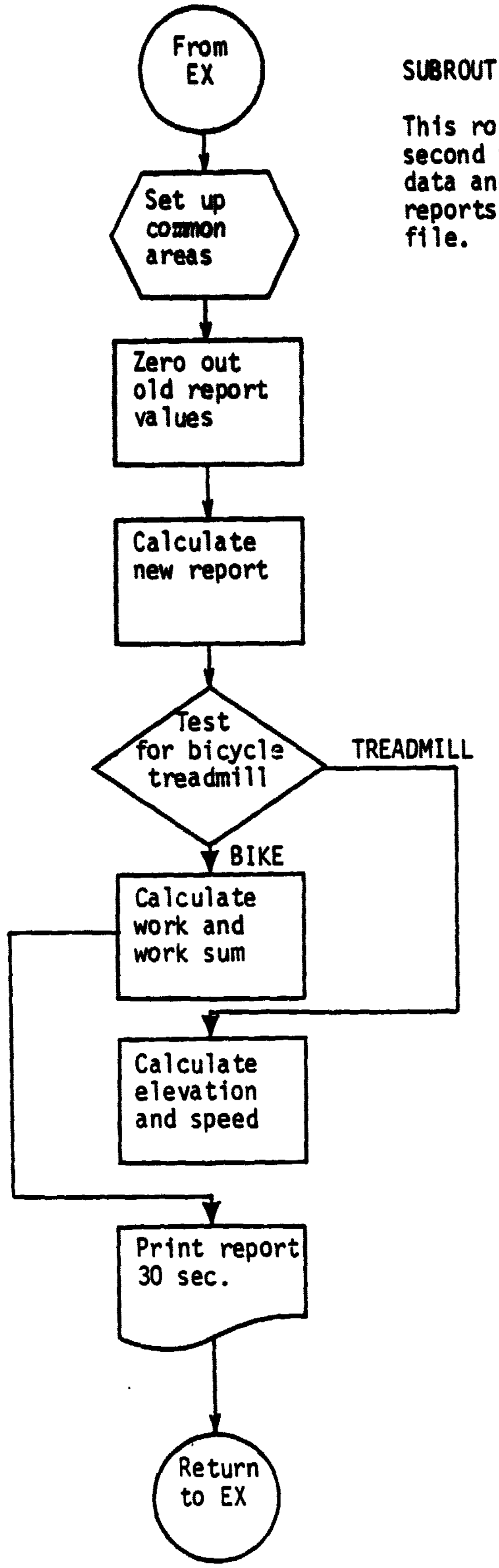
SUBROUTINE TRSBUF

This subroutine disables the clock interrupts by altering the program status word then makes a copy of ADCOM1 and GAS30. ADCOM1 and GAS30 are zeroed and certain values in ADCOM1 are reset in their original value. The clock interrupts are again enabled.


From EX
Set up common areas
Get end of breath pointer
Calculate the average for O_2, N_2, CO_2
Normalize the gases
Calculate sum O_2 consumption & CO_2 production
Test for another breath
YES
NO
Return to EX

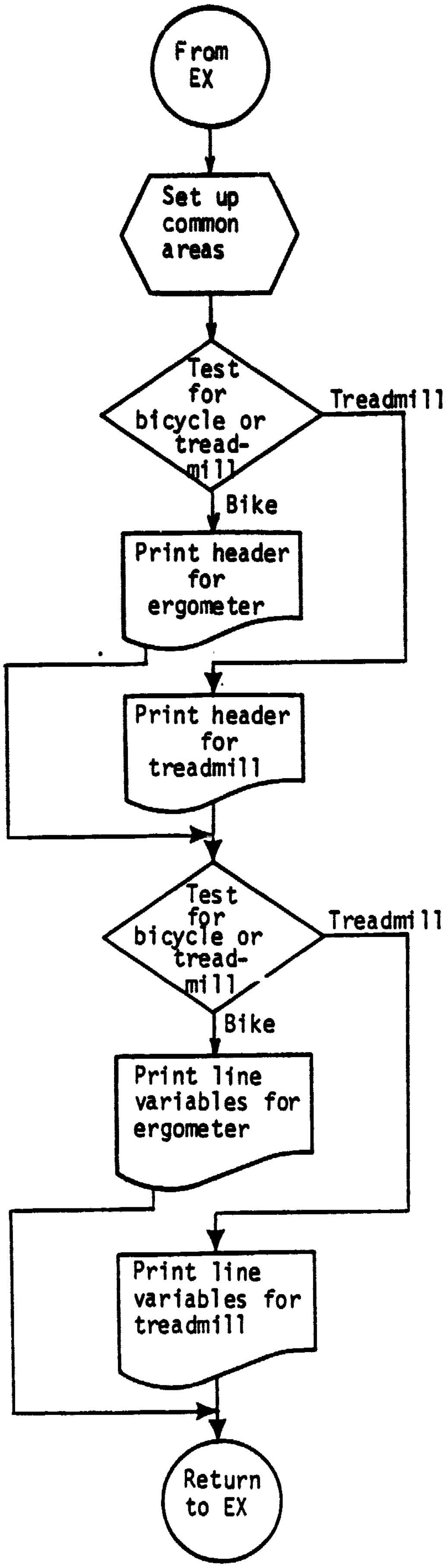
SUBROUTINE BREATH

Subroutine BREATH is used to determine the oxygen consumption and carbon dioxide production in each breath exhaled by the subject and keep a 30 sec. sum.


From EX
Set up common areas
Zero out old report values
Calculate new report
Test for bicycle treadmill
TREADMILL
BIKE
Calculate work and work sum
Calculate elevation and speed
Print report 30 sec.
Return to EX

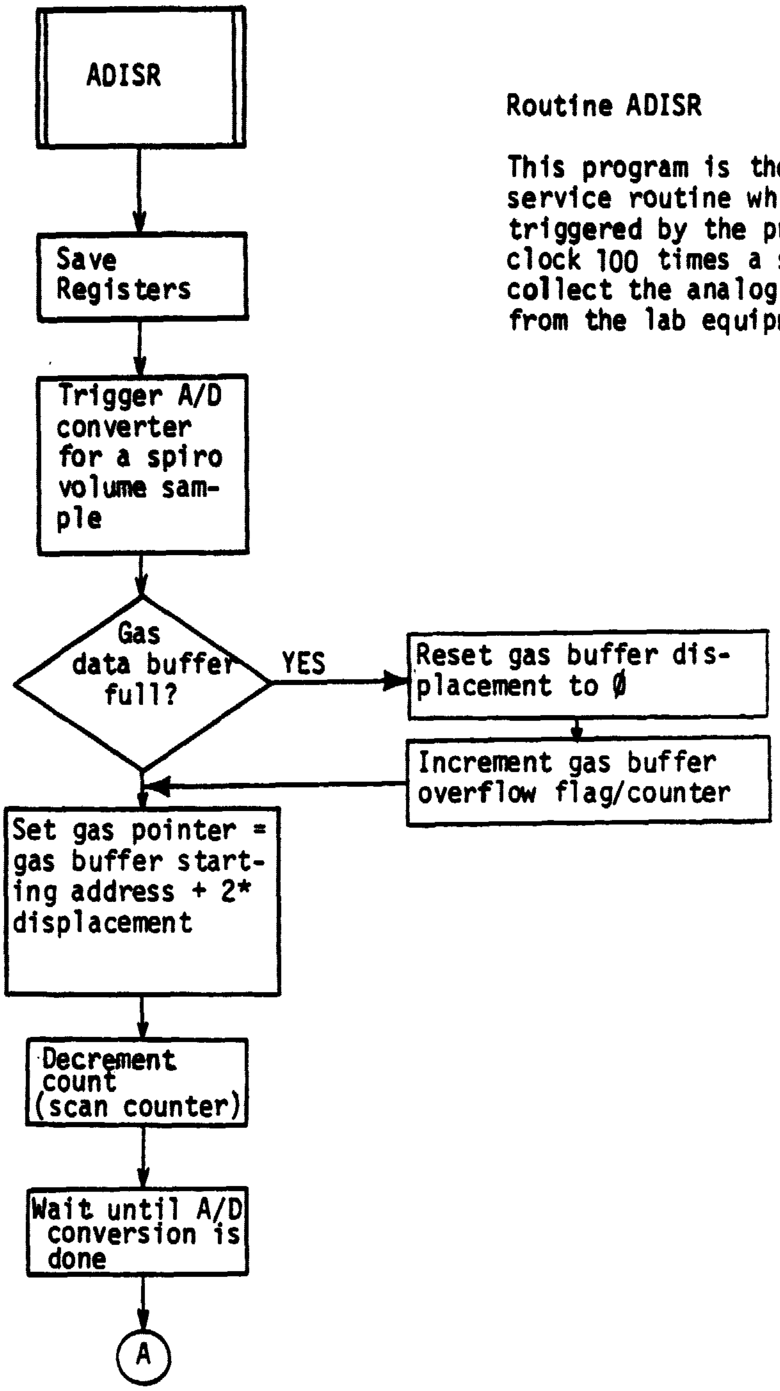
SUBROUTINE RPT30

This routine prints the 30 second reports for the collected data and places the 1 minute reports in the patients data file.


From EX
Set up common areas
Test for bicycle or tread-mill
Treadmill
Bike
Print header for ergometer
Print header for treadmill
Test for bicycle or tread-mill
Treadmill
Bike
Print line variables for ergometer
Print line variables for treadmill
Return to EX

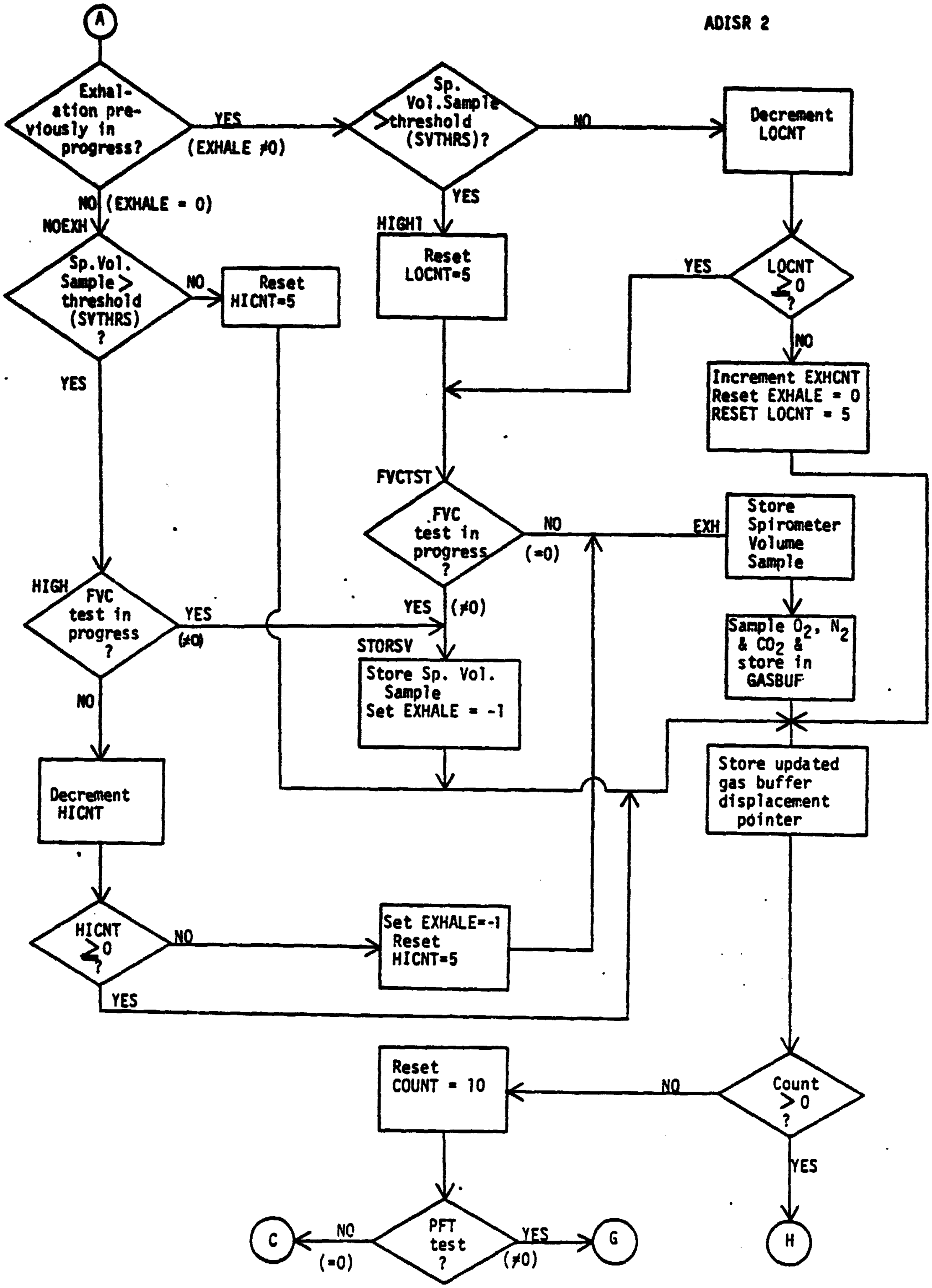
SUBROUTINE REPORT

Prints the correct headers for the 30 sec reports depending if test is a treadmill or bicycle.

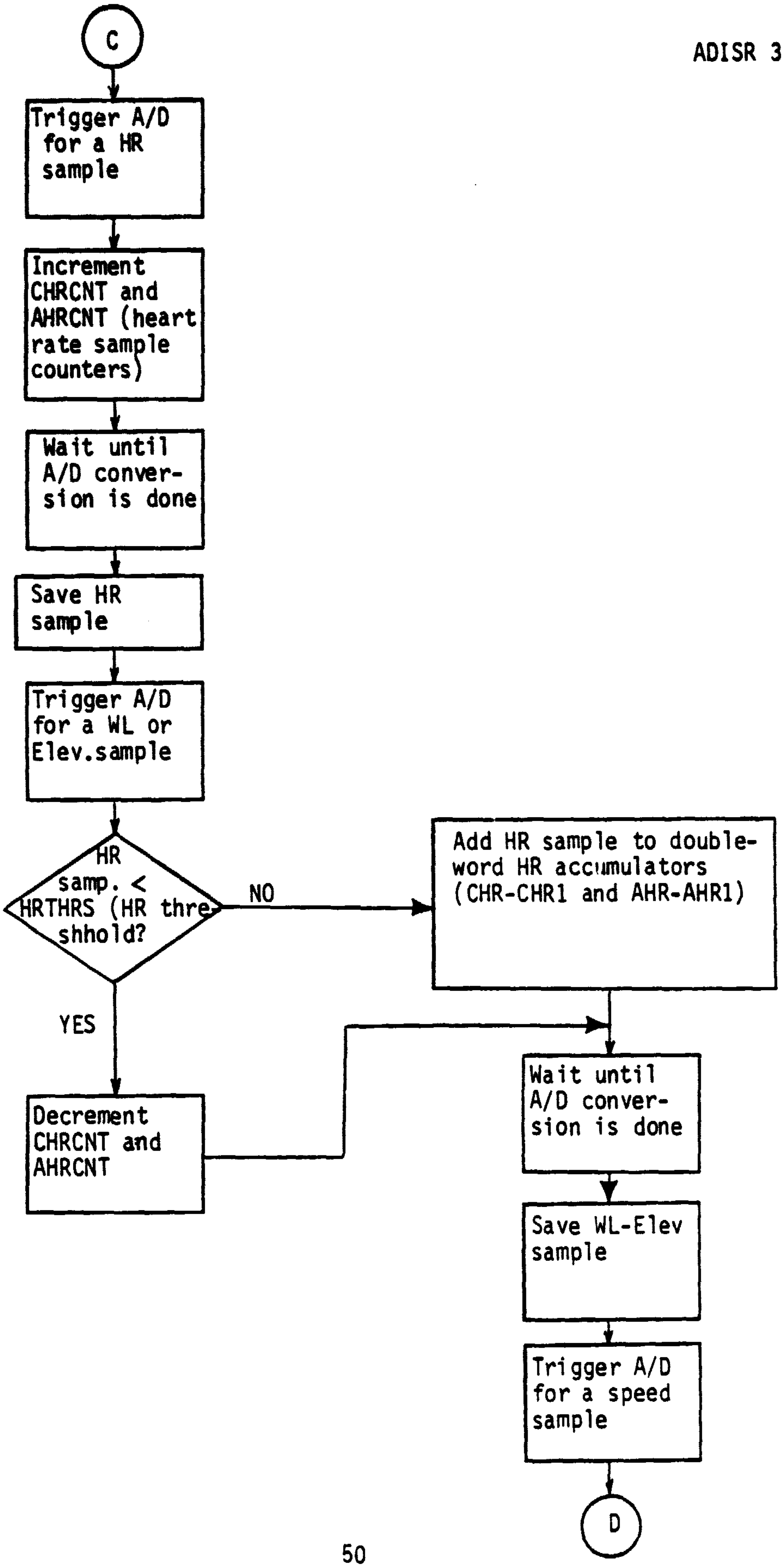

ADISR
Save Registers
Trigger A/D converter for a spiro volume sample
Gas data buffer full?
YES
Reset gas buffer displacement to Ø
Increment gas buffer overflow flag/counter
Set gas pointer = gas buffer starting address + 2* displacement
Decrement count (scan counter)
Wait until A/D conversion is done
A

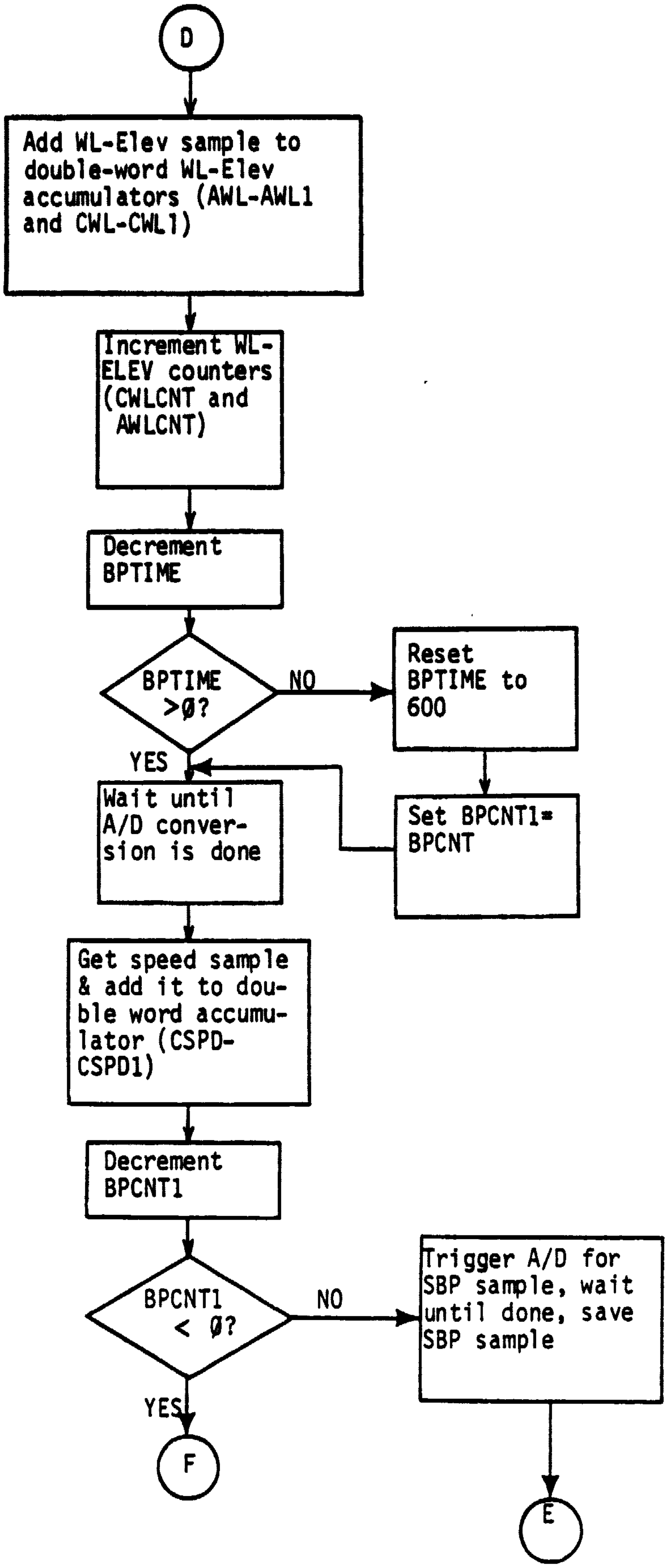
Routine ADISR

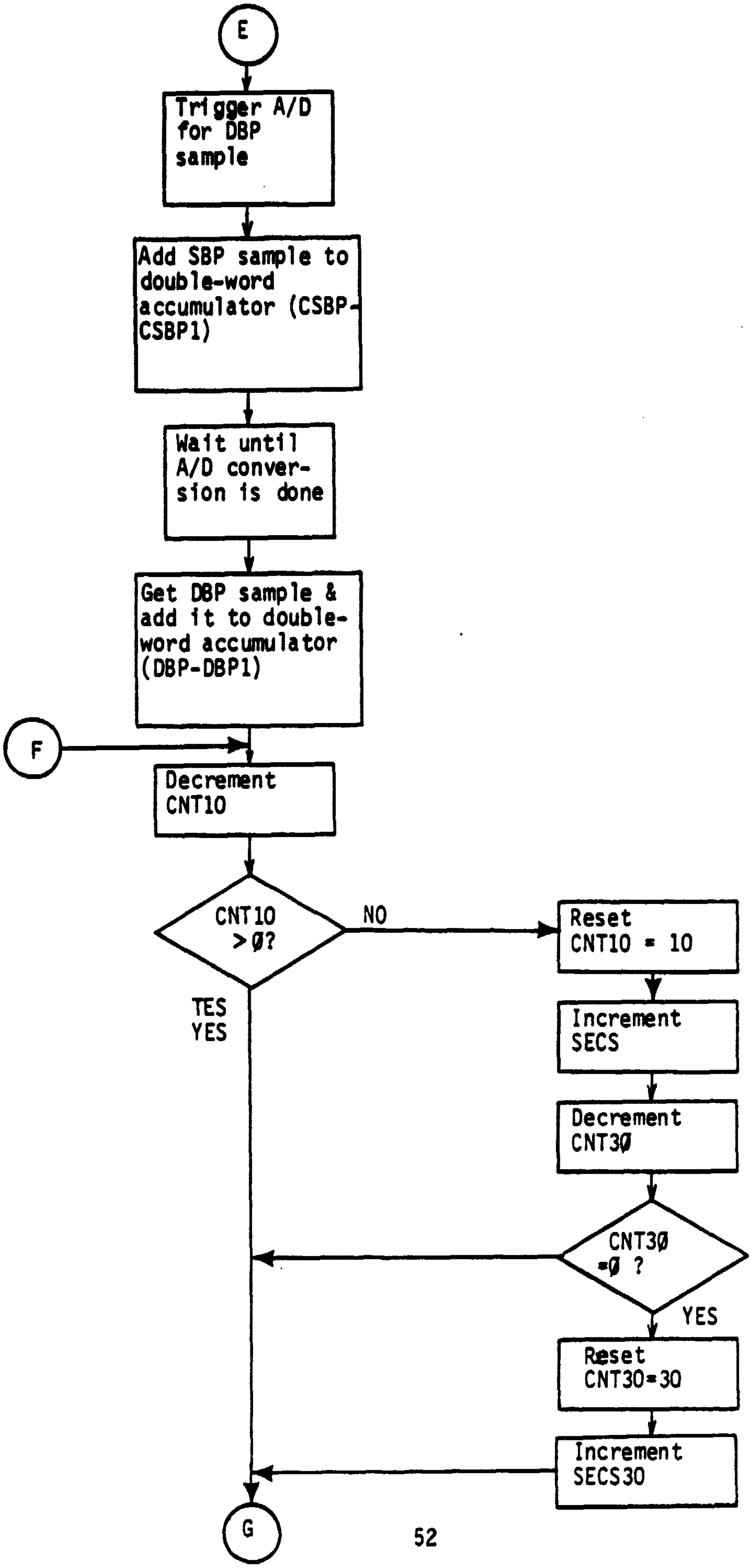
This program is the interrupt service routine which is triggered by the programmable clock 100 times a second to collect the analog signals from the lab equipment.

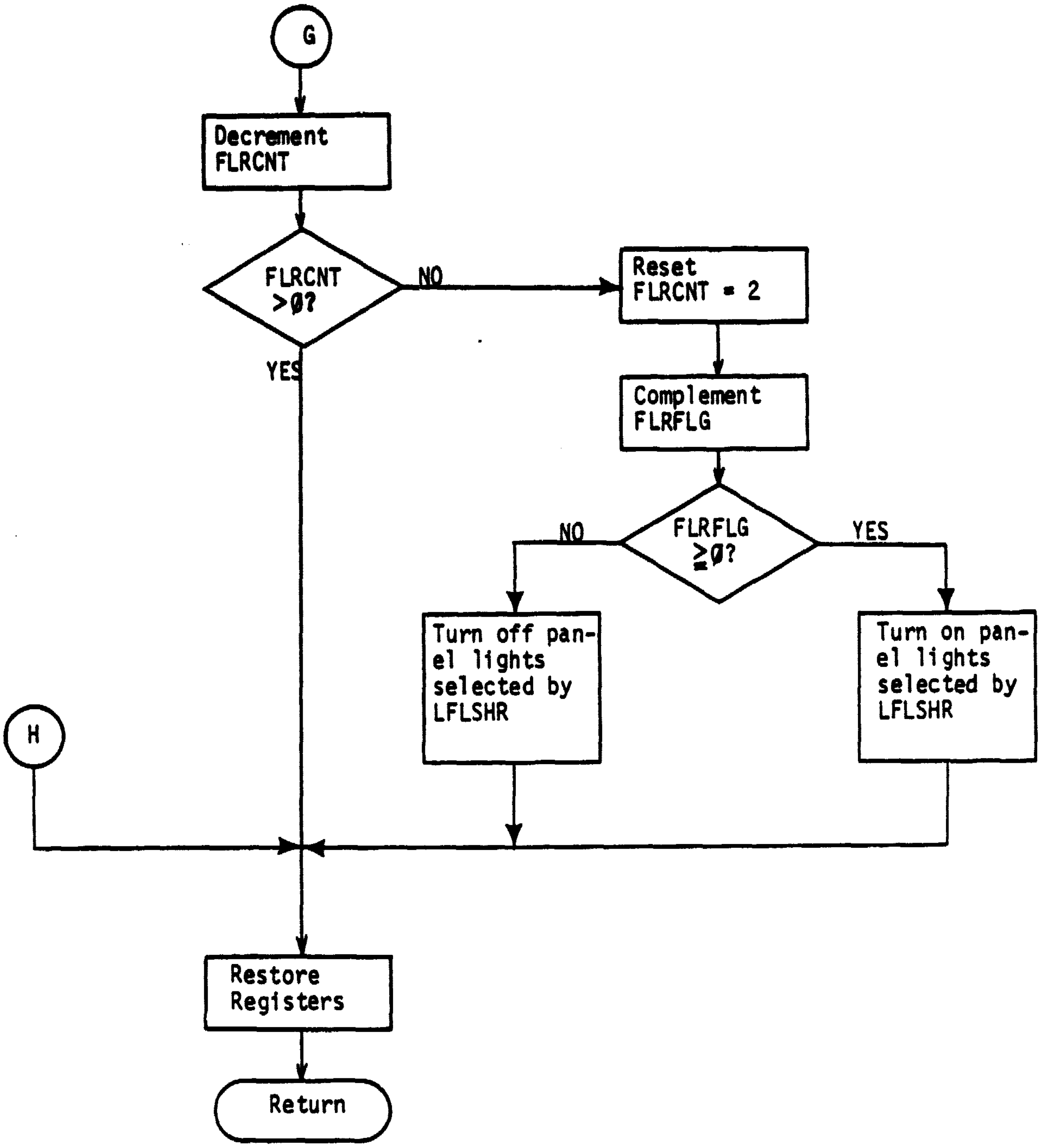

A
Exhalation previously in progress?
YES
(EXHALE ≠0)
NO
(EXHALE = 0)
NOEXH
Sp. Vol. Sample > threshold (SVTHRS)?
NO
YES
HIGH1
Reset LOCNT=5
Decrement LOCNT
LOCNT ≥ 0 ?
YES
NO
Increment EXHCNT
Reset EXHALE = 0
RESET LOCNT = 5
Sp. Vol. Sample > threshold (SVTHRS) ?
NO
Reset HICNT=5
YES
FVCTST
FVC test in progress ?
NO
(=0)
EXH
Store Spirometer Volume Sample
Sample O_2, N_2 & CO_2 & store in GASBUF
HIGH
FVC test in progress ?
YES
(≠0)
YES
(≠0)
STORSV
Store Sp. Vol. Sample
Set EXHALE = -1
NO
Decrement HICNT
Store updated gas buffer displacement pointer
HICNT ≥ 0 ?
NO
Set EXHALE=-1
Reset HICNT=5
YES
Reset COUNT = 10
NO
Count > 0 ?
YES
PFT test ?
NO
(=0)
C
YES
(≠0)
G
H

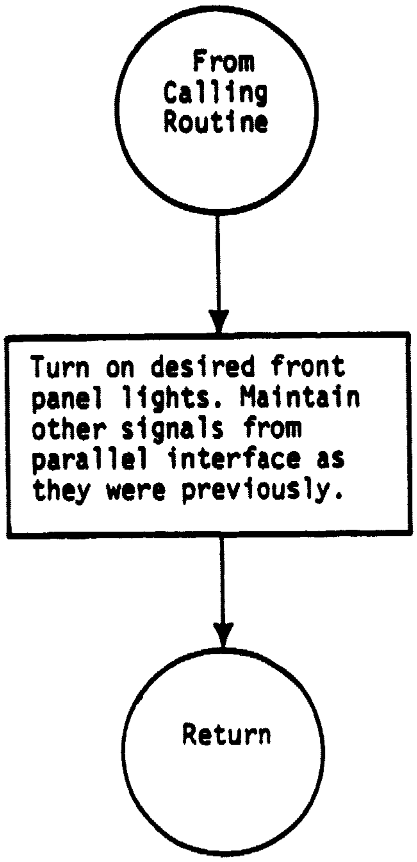
ADISR 3


C
Trigger A/D for a HR sample
Increment CHRCNT and AHRCNT (heart rate sample counters)
Wait until A/D conversion is done
Save HR sample
Trigger A/D for a WL or Elev.sample
HR samp. < HRTHRS (HR threshhold?
NO
Add HR sample to double-word HR accumulators (CHR-CHR1 and AHR-AHR1)
YES
Decrement CHRCNT and AHRCNT
Wait until A/D conversion is done
Save WL-Elev sample
Trigger A/D for a speed sample
D


D
Add WL-Elev sample to double-word WL-Elev accumulators (AWL-AWL1 and CWL-CWL1)
Increment WL-ELEV counters (CWLCNT and AWLCNT)
Decrement BPTIME
BPTIME >Ø?
NO
Reset BPTIME to 600
YES
Set BPCNT1= BPCNT
Wait until A/D conversion is done
Get speed sample & add it to double word accumulator (CSPD-CSPD1)
Decrement BPCNT1
BPCNT1 < Ø?
NO
Trigger A/D for SBP sample, wait until done, save SBP sample
YES
F
E

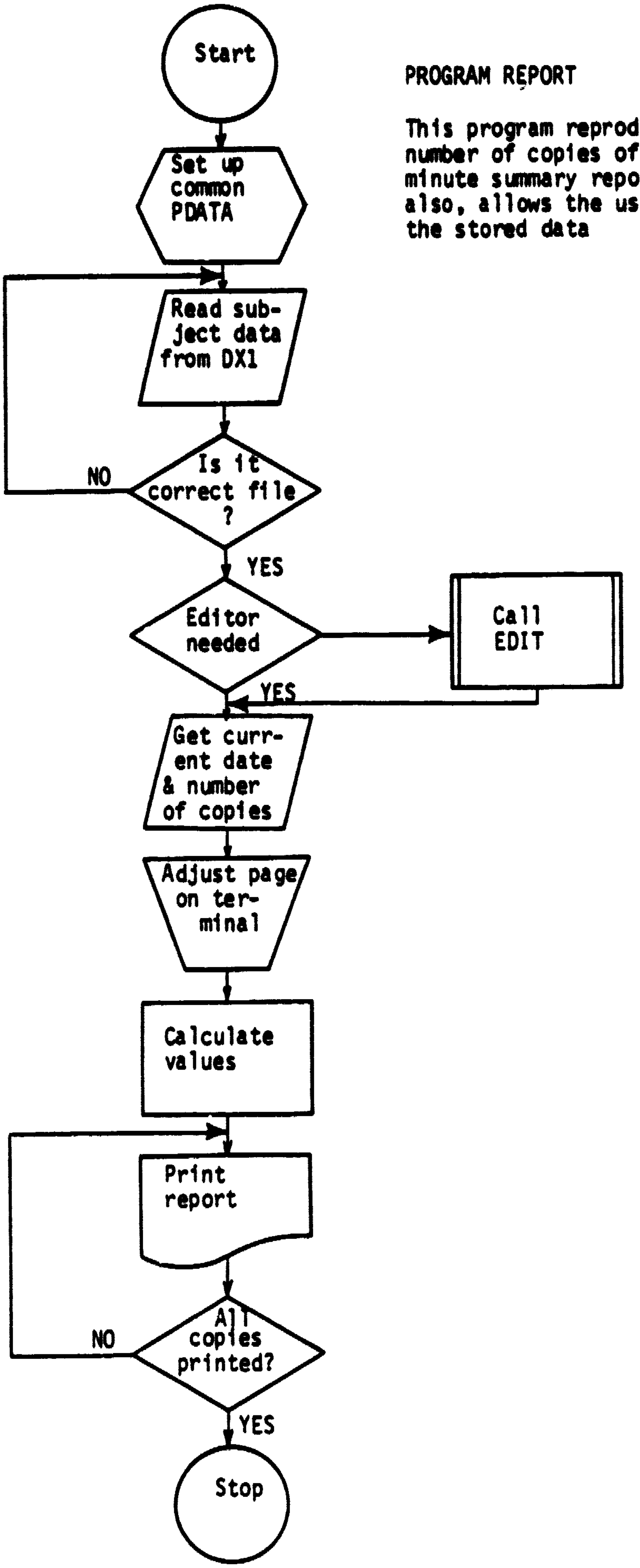

E
Trigger A/D for DBP sample
Add SBP sample to double-word accumulator (CSBP-CSBP1)
Wait until A/D conversion is done
Get DBP sample & add it to double-word accumulator (DBP-DBP1)
F
Decrement CNT10
CNT10 >Ø?
NO
Reset CNT10 = 10
TES
YES
Increment SECS
Decrement CNT3Ø
CNT3Ø =Ø ?
YES
Reset CNT30=30
Increment SECS30
G


G
Decrement FLRCNT
FLRCNT >Ø?
NO
YES
Reset FLRCNT = 2
Complement FLRFLG
FLRFLG ≥Ø?
NO
YES
Turn off panel lights selected by LFLSHR
Turn on panel lights selected by LFLSHR
H
Restore Registers
Return


From Calling Routine
Turn on desired front panel lights. Maintain other signals from parallel interface as they were previously.
Return

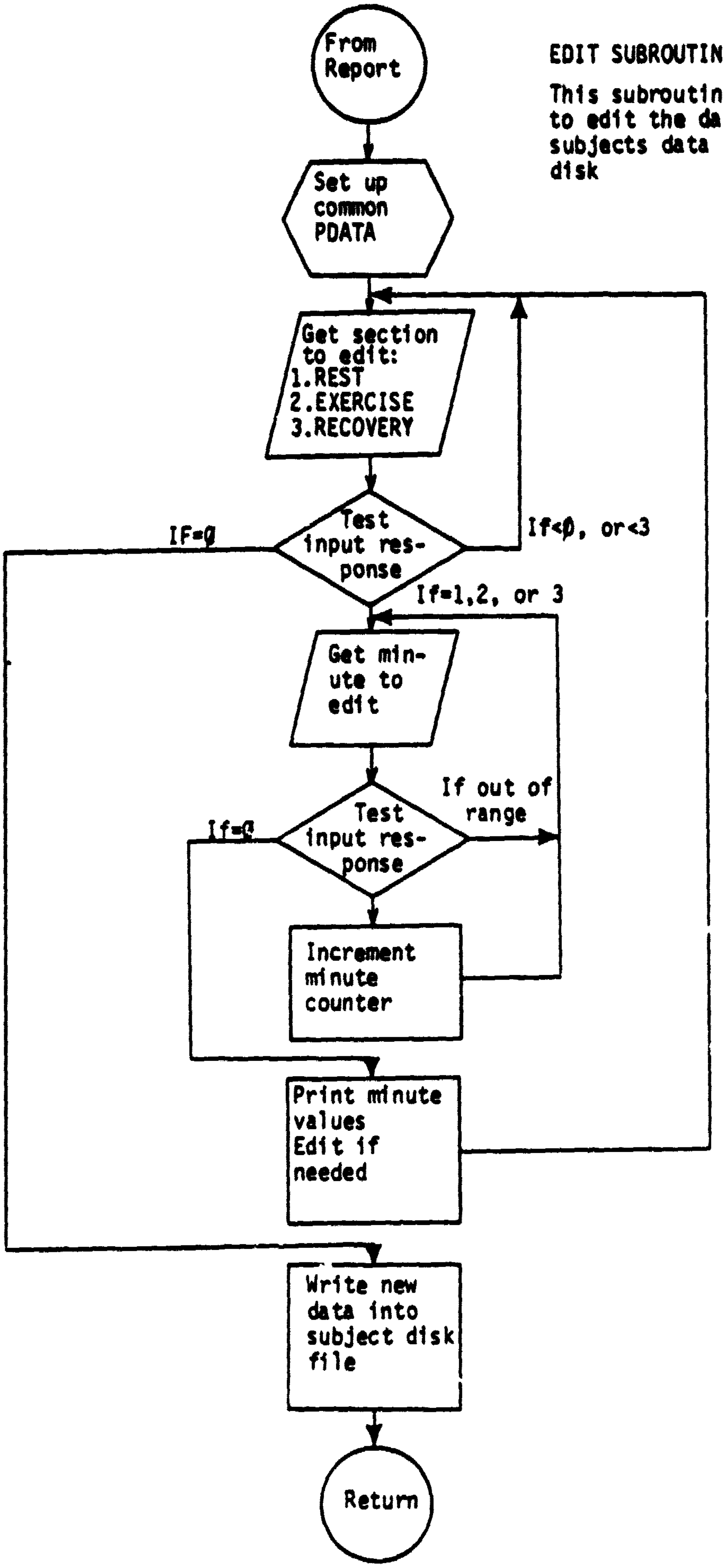
SUBROUTINE LIGHT

This routine is used by subroutine CAL and EX to turn on the appropriate paneī lights to cue the operator to the functions allowed at various times during a test.



PROGRAM REPORT

This program reproduces n number of copies of the one minute summary report; and also, allows the user to edit the stored data

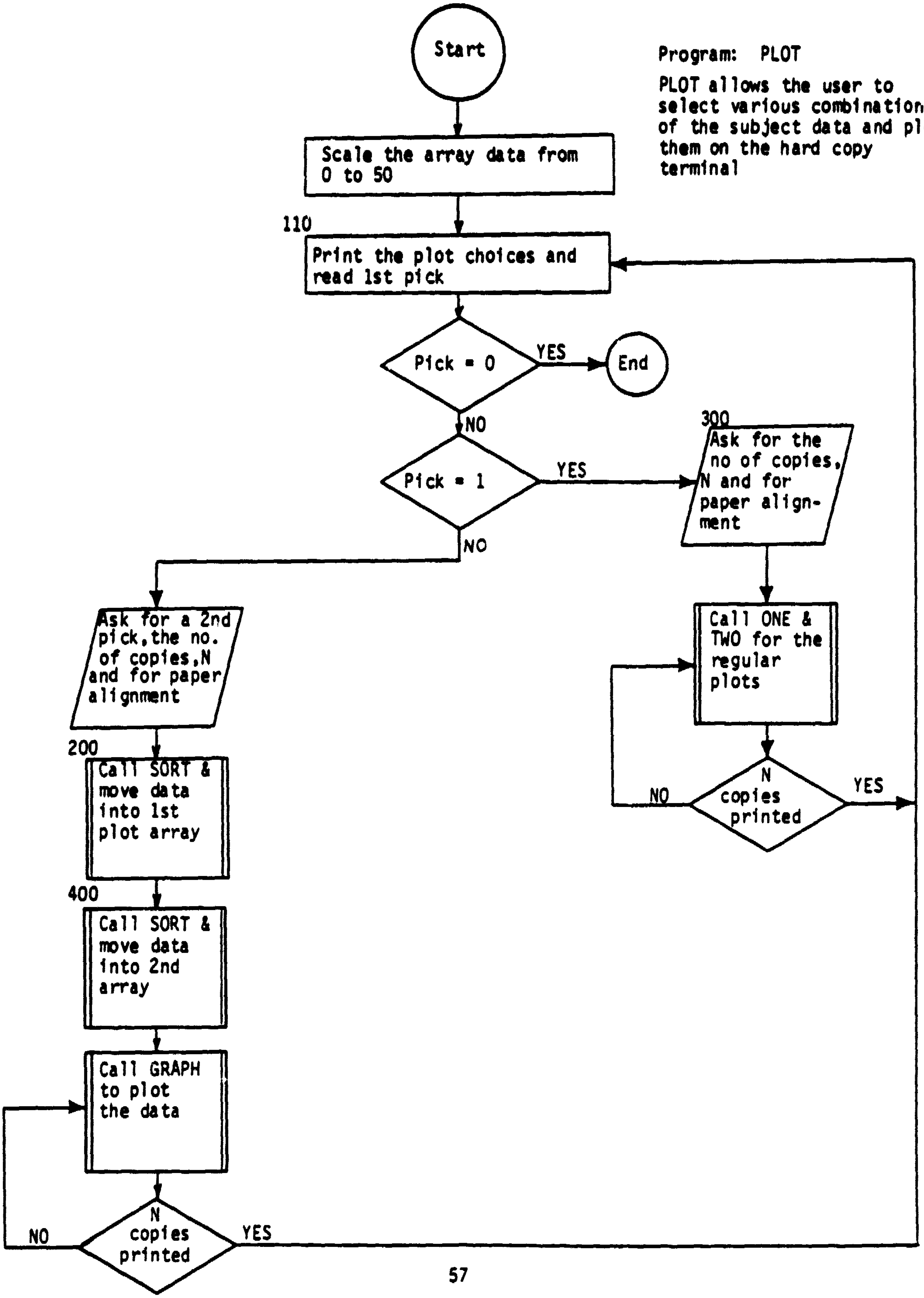


EDIT SUBROUTINE

This subroutine allows the user to edit the data stored in the subjects data file stored on disk

Program: PLOT

PLOT allows the user to select various combinations of the subject data and plot them on the hard copy terminal


Start
Scale the array data from 0 to 50
110
Print the plot choices and read 1st pick
Pick = 0
YES
End
NO
Pick = 1
YES
300
Ask for the no of copies, N and for paper alignment
Call ONE & TWO for the regular plots
N copies printed
NO
YES
NO
Ask for a 2nd pick, the no. of copies, N and for paper alignment
200
Call SORT & move data into 1st plot array
400
Call SORT & move data into 2nd array
Call GRAPH to plot the data
N copies printed
NO
YES

APPENDIX H

PROGRAM LISTINGS

```
C  **************************************************************
C  **************************************************************
C  *                                                            *
C  *                 TECHNOLOGY INCORPORATED                    *
C  *                  LIFE SCIENCE DIVISION                     *
C  *                                                            *
C  **************************************************************
C  *                                                            *
C  *  PROGRAM NAME:............... PRETES                       *
C  *  AUTHOR:..................... DOROTHY STEPHENS             *
C  *  DATE:....................... 2/NOV/78                     *
C  *  MODIFIED:................... ABHIJIT GADGIL (DEC/1979)    *
C  *                                                            *
C  *------------------------------------------------------------*
C  *                                                            *
C  *  COMPUTER SYSTEM:........... DEC PDP 1103 "LSI-11"         *
C  *  OPERATING SYSTEM:........... RT-11  V01C-03A              *
C  *                                                            *
C  *------------------------------------------------------------*
C  *                                                            *
C  *  COMPILING SEQUENCE:                                       *
C  *                                                            *
C  *     .R FORTRA <CR>                                         *
C  *     *PRETES=PRETES/L <CR>                                  *
C  *     *<CR>                                                  *
C  *                                                            *
C  *------------------------------------------------------------*
C  *                                                            *
C  *  RUN MODUAL LINKING SEQUENCE:                              *
C  *                                                            *
C  *     .R LINK <CR>                                           *
C  *     *PRETES=PRETES,DX0:SYSLIB/F <CR>                       *
C  *     *<CR>                                                  *
C  *                                                            *
C  *------------------------------------------------------------*
C  *                                                            *
C  *  CALLING SEQUENCE:                                         *
C  *                                                            *
C  *     .R PRETEST <CR>                                        *
C  *                                                            *
C  *------------------------------------------------------------*
C  *                                                            *
C  *  PURPOSE:                                                  *
C  *                                                            *
C  *     ALLOWS THE USER TO CREATE A PATIENT DATA FILE ON DX1:  *
C  *     AND ENTER IN THE PERTINENT SUBJECT INFORMATION.        *
C  *                                                            *
C  **************************************************************
C  **************************************************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
        C
        C          PROGRAM: PRETEST
        C
        C          PURPOSE: QUESTIONS NECESSARY FOR TREADMILL
        C
        C                   AND BICYCLE SUBJECT DATA
        C
        C
        C
        C*************************************************************
0001          REAL CRT(10)
0002          LOGICAL*1 YES,JNUM
0003          REAL*4 IMONTH,MONTH(12)
0004          DATA MONTH/'JAN','FEB','MAR','APR','MAY','JUN','JUL',
             %'AUG','SEP','OCT','NOV','DEC'/
0005          COMMON/X/MON(24)
0006          INTEGER NFILE(3),FILE(4),ICRT(1536)
0007          EQUIVALENCE (ICRT(50),CRT(1))
0008          EQUIVALENCE (ICRT(54),TMP),(ICRT(57),PRS)
0009          DATA FILE/3RDX1,3RXXX,3RXXX,3RDAT/
0010          DATA YES/1HY/
        C  ZERO OUT THE DATA BLOCKS (6 OF THEM)
0011          DO 800 I=1,1536
0012    800   ICRT(I)=0
        C * * * * *  SET UP QUEUE ELEMENTS FOR DISK I/O
0013          CALL LOCK
0014           IF(IQSET(3).NE. 0) STOP 'QUEUE ELEMENT FAILURE'
0016          IGOOF=0
0017            TYPE 400
0018    400    FORMAT(' PATIENT FILENAME :',$)
0019           ACCEPT 410,NFILE
0020    410    FORMAT(3A2)
0021           CALL IRAD50(6,NFILE,FILE(2))
        C * * * *  OPENING DISK FILE
0022           ICHAN=IGETC()
0023           IF(ICHAN .LT.0) STOP 'NO AVAILABLE CHANNEL'
0025           IF(IENTER(ICHAN,FILE,6).LT. 0) STOP 'FILE ALLOCATION FAILURE'
0027          TYPE 199
0028    199     FORMAT('                 PRETEST QUESTIONS ')
0029          IF(IGOOF.GT.0) GO TO 121
0031    521       TYPE 21
0032    21     FORMAT(' 1. SUBJECT NAME: ',$)
0033           ACCEPT 22, (ICRT(I),I=26,39)
0034    22     FORMAT(14A2)
0035          IF(IGOOF .GT. 0) GO TO 121
0037    522      TYPE 1
0038    1       FORMAT(' 2. SUBJECT SEX(M=MALE,F=FEMALE): ',$)
0039           ACCEPT 2, ICRT(2)
0040    2      FORMAT(A2)
0041          IF(IGOOF.GT.0) GO TO 121
0043    523       TYPE 23
0044    23       FORMAT(' 3. SOCIAL SEC. NO.(NNN NN NNNN) ')
0045           ACCEPT 3, (ICRT(I),I=3,5)
0046    3      FORMAT(I3,1X,I2,1X,I4)
```

```
        IF(IGOOF.GT.0) GO TC 121
524         TYPE 33
           FORMAT(' 4. CATE OF BIFTH (DC-MMM-YY): ')
         ACCEPT 4,IDAY,IMONTH,IYR
4        FORMAT( I2,1X,A3,1X,I2)
        DO 100 I=1,12
        IF(IMONTH.NE.MCNTH(I)) GC TC 100
        ICRT(22)=I
        ICRT(23)=IDAY
        ICRT(24)=IYR
        GO TO 300
100     CONTINUE
        TYPE 200
200     FORMAT(' ERROR IN DATE OF BIRTH. PLEASE RE-ENTER')
        GO TO 524
300     CONTINUE
        IF(IGOOF.GT.0) GO TC 555
525         TYPE 65
65         FORMAT(' 5. HEIGHT (CM): ',$)
         ACCEPT 5, CRT(1)
 5       FORMAT(F6.0)
        IF(IGOOF.GT.0) GO TO 121
526        TYPE 266
266        FORMAT(' 6. WEIGHT (KG) :',$)
         ACCEPT 6, CRT(2)
 6        FORMAT(F6.1)
        IF(IGOOF.GT.0) GO TC 121
527         TYPE 67
67         FORMAT(' 7. TEST DATE (CC-MMM-YY):')
         ACCEPT 7, IDAY,IMCNTH,IYR
7        FORMAT(I2,1X,A3,1X,I2)
        DO 500 I=1,12
        IF(IMONTH.NE.MCNTH(I)) GC TC 500
        ICRT(40)=I
        ICRT(41)=IDAY
        ICRT(42)=IYR
        GO TO 700
500     CONTINUE
        TYPE 600
600     FORMAT(' ERROR IN TEST DATE. PLEASE RE-ENTER')
        GO TO 527
700     CONTINUE
        GO TO 555
31       TYPE 68
68        FORMAT(' 8. UNIGUE NO.(AANNNN): ')
         ACCEPT 8,ICRT(7), ICRT(8)
8         FORMAT(A2,I4)
        IF(IGOOF.GT.0) GO TC 121
528         TYPE 69
69       FORMAT(' 9. RETEST NO.: ',$)
         ACCEPT 9, ICRT(9)
9        FORMAT(I2)
        IF(IGCOF.GT.0) GC TC 121
529         TYPE 70
```

```
0109  70        FORMAT(' 10. CPID NO.(NNN): ')
0110           ACCEPT 10, ICRT(10)
0111  10       FORMAT(I4)
0112          IF(IGOOF.GT.0) GO TO 121
0114  530          TYPE 71
0115  71        FORMAT(' 11. TEST MODE:(1=LIVE,2=TAPE) ',$)
0116           ACCEPT 11, ICRT(11)
0117  11       FORMAT(I2)
0118          IF(IGOOF.GT.0) GO TO 121
0120  531          TYPE 72
0121  72       FORMAT (' 12. TYPE OF TEST : (2=TREADMILL,1=ERGOMETER) ',$)
0122           ACCEPT 12, ICRT(12)
0123  12       FORMAT(I2)
0124          IF(IGOOF.GT.0) GO TO 121
0126  532          TYPE 73
0127  73       FORMAT(' 13. AMBIENT TEMP.(C) : ',$)
0128           ACCEPT 13, TMP
0129  13        FORMAT(F5.0)
0130          IF(IGOOF.GT.0) GO TO 121
0132  533          TYPE 74
0133  74       FORMAT(' 14. AMBIENT PRESSURE(MMHG): ',$)
0134           ACCEPT 14, PRS
0135  14      FORMAT(F6.0)
0136          IF(IGOOF.GT.0) GO TO 121
0138          IF(ICRT(12) .EQ. 2) GO TO 121
0140  534          TYPE 75
0141  75      FORMAT(' 15. TARGET HR @ 4 MIN.(BPM): ',$)
0142           ACCEPT 276, ICRT(15)
0143  276      FORMAT(I4)
0144          IF(IGOOF.GT.0) GO TO 121
0146  535          TYPE 277
0147  277      FORMAT(' 16. TARGET HR @ 8 MIN.(BPM): ',$)
0148           ACCEPT 278,ICRT(16)
0149  278      FORMAT(I4)
0150          IF(IGOOF.GT.0) GO TO 121
0152  536          TYPE 279
0153  279      FORMAT(' 17. TARGET HR @ 12 MIN.(BPM): ',$)
0154           ACCEPT 280,ICRT(17)
0155  280      FORMAT(I4)
0156          IF(IGOOF.GT.0) GO TO 121
0158  537          TYPE 281
0159  281      FORMAT(' 18. TARGET HR @ 16 MIN.(BPM): ',$)
0160           ACCEPT 282,ICRT(18)
0161  282     FORMAT(I4)
0162          IF(IGOOF.GT.0) GO TO 121
0164  538         TYPE 191
0165  191       FORMAT(' 19. MAX TEST ? (1=YES,2=NO): ',$)
0166                ACCEPT 19,ICRT(19)
0167  19       FORMAT(I2)
0168          IF(ICRT(19) .EQ. 2) GO TO 121
0170          ICRT(20)=220-ICRT(25)
0171           TYPE 20,ICRT(20)
0172  20       FORMAT(' 20. TARGET HR @ MAX: ',I4)
0173  50       FORMAT(I2)
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:01:22                PAGE 004

0174  121        TYPE 122
0175  122       FORMAT(' ALL INFORMATICN CORRECT ? (Y=YES) ')
0176           ACCEPT 222,JNUM
0177  222          FORMAT(A1)
0178            IF(JNUM .EQ. YES) GO TO 120
0180           IGOUF=IGOOF+1
0181            TYPE 322
0182  322       FORMAT(' TYPE QUESTION NUMBER OF INCORRECT ENTRY ')
0183            ACCEPT 323,INUM
0184  323       FORMAT(I2)
0185           IF(INUM .LE. 0) GO TO 338
0187            IF(INUM .GT.20) GO TO 338
0189            GO TO 490
0190  338       TYPE 339
0191  339       FORMAT(' * * * * * TRY AGAIN, WRONG ENTRY ! ! ! ! ! ')
0192           GO TO 121
0193  490        GO TO (521,522,523,524,525,526,527,31,528,529,530
             *,531,532,533,534,535,536,537,538),INUM
      C
      C          CALCULATE AGE FROM DOB AND TEST DATE
      C
      C
      C
      C            MONTH CHECK * * * * * * * * * *
      C
      C
0194  555      ICRT(25)=ICRT(42)-ICRT(24)
0195           IF (ICRT(22)-ICRT(40)) 666,664,663
0196  663      ICRT(25)=ICRT(25)-1
0197           GO TO 666
0198  664      IF (ICRT(23).GT.ICRT(41)) GO TO 663
0200  666        TYPE 667, ICRT(25)
0201  667         FORMAT(' AGE: ',I2)
0202           IF(IGUOF.GT.0) GO TO 121
0204           GO TO 31
      C * * * * * * * * *  WRITING TO DISK FILE
0205  120        IF(IWRITW(1536,ICRT,0,ICHAN).LT.0) STOP 'DISK WRITE ERROR'
      C * * * * * *  CLOSING DISK FILE
0207           CALL UNLOCK
0208             CALL CLOSEC(ICHAN)
0209             STOP
0210             END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

CRT      000150  REAL*4     ARRAY (10)
MONTH    006006  REAL*4     ARRAY (12)
NFILE    006066  INTEGER*2 ARRAY (3)
FILE     006074  INTEGER*2 ARRAY (4)
ICRT     000006  INTEGER*2 ARRAY (1536)
YES      006104  LOGICAL*1 VARIABLE
JNUM     010202  LOGICAL*1 VARIABLE
IMONTH   010204  REAL*4     VARIABLE
TMP      000160  REAL*4     VARIABLE
PRS      000166  REAL*4     VARIABLE
I        010210  INTEGER*2 VARIABLE
LOCK     000000  INTEGER*2 PROCEDURE
IQSET    000000  INTEGER*2 PROCEDURE
IGOOF    010212  INTEGER*2 VARIABLE
IRAD50   000000  INTEGER*2 PROCEDURE
ICHAN    010214  INTEGER*2 VARIABLE
IGETC    000000  INTEGER*2 PROCEDURE
IENTER   000000  INTEGER*2 PROCEDURE
IDAY     010216  INTEGER*2 VARIABLE
IYR      010220  INTEGER*2 VARIABLE
INUM     010222  INTEGER*2 VARIABLE
IWRITW   000000  INTEGER*2 PROCEDURE
UNLOCK   000000  REAL*4     PROCEDURE
CLOSEC   000000  REAL*4     PROCEDURE


COMMON BLOCK /X/         LENGTH 000060

MON      000000  INTEGER*2 ARRAY (24)
```

```
C  *************************************************************
C  *************************************************************
C  *                                                           *
C  *                 TECHNOLOGY INCORPORATED                   *
C  *                  LIFE SCIENCE DIVISION                    *
C  *                                                           *
C  *************************************************************
C  *                                                           *
C  *  PROGRAM NAME:............... STRESS                      *
C  *  AUTHOR:..................... ROY A. REED                 *
C  *         ..................... CHUCK MANN                  *
C  *         ..................... WILLIAM G. CROSIER          *
C  *  DATE:....................... 2/NOV/78                    *
C  *                                                           *
C  *-----------------------------------------------------------*
C  *                                                           *
C  *  COMPUTER SYSTEM:.......... DEC PDP 1103 "LSI-11"         *
C  *  OPERATING SYSTEM:......... RT-11  V01C-03A               *
C  *                                                           *
C  *-----------------------------------------------------------*
C  *                                                           *
C  *  COMPILING SEQUENCE:                                      *
C  *                                                           *
C  *     .R FORTRA <CR>                                        *
C  *     *STRESS=STRESS,INITI,DATA/U <CR>                      *
C  *     *CAL=CAL <CR>                                         *
C  *     *PFT=PFT <CR>                                         *
C  *     *EX=EX,RPTHDR,WRKADJ,BREATH,RPT30,TIME <CR>           *
C  *     *LIGHT=LIGHT <CR>                                     *
C  *     *<CR>                                                 *
C  *     .R MACRO <CR>                                         *
C  *     *ADISR=ADISR <CR>                                     *
C  *     *TRSBUF=TRSBUF <CR>                                   *
C  *     *SNATCH=SNATCH<CR>                                    *
C  *     * <CR>                                                *
C  *                                                           *
C  *-----------------------------------------------------------*
C  *                                                           *
C  *  RUN MODUAL LINKING SEQUENCE:                             *
C  *                                                           *
C  *     .R LINK <CR>                                          *
C  *     *STRESS=STRESS,CAL,PFT,EX,DX0:SYSLIB/F/C <CR>         *
C  *     *LIGHT,ADISR,TRSBUF,SNATCH <CR>                       *
C  *     *<CR>                                                 *
C  *                                                           *
C  *-----------------------------------------------------------*
C  *                                                           *
C  *  CALLING SEQUENCE:                                        *
C  *                                                           *
C  *     .R STRESS <CR>                                        *
C  *                                                           *
C  *-----------------------------------------------------------*
C  *                                                           *
C  *  PURPOSE:                                                 *
C  *                                                           *
C  *     A MICROCOMPUTER - BASED DATA ACQUISITION AND CONTROL  *
C  *     SYSTEM FOR CARDIOPULMONARY STRESS TESTING.            *
C  *                                                           *
C  *************************************************************
C  *************************************************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
        C      PROGRAM:   PROCESS, VERSION 2.0     29 AUG. 1979
        C      FUNCTION:  RUN CARDIOPULMONARY STRESS STEST
        C
0001           INTEGER MAJREV,MINREV
0002           INTEGER DROUTB,DRINB,TESTYP
0003           INTEGER*4 IN,OUT
0004           COMMON /PDATA/IDATA(1536)
0005           DATA IN/'IN  '/,OUT/'OUT '/
0006           COMMON/X/MO(24)
0007           DATA MO/'JA','N ','FE','B ','MA','R ','AP','R ',
             @'MA','Y ','JU','N ','JU','L ','AU','G ','SE','P ',
             @'OC','T ','NO','V ','DE','C '/
0008           EQUIVALENCE (IDATA(13),MAJREV),(IDATA(14),MINREV)
        C--    SETS THE MAJOR AND MINOR REVISION NUMBERS
        C      FOR THIS VERSION OF CVDAS
0009           MAJREV=2
0010           MINREV=0
        C
        C--DRV11 PARALLEL INTERFACE ADDRESSES:
0011           DROUTB= "167772         ! "  OUTPUT BUFFER (KATHY'S DRAI,DRBI)
0012           DRINB= "167774          ! "  INPUT BUFFER (KATHY'S DRAO,DRBO)
        C     THE LOW ORDER TWO BITS OF DRCSR  WILL BE DINGLED
        C      BY PROCES FOR CALIBRATION
        C
        C
        C--LOCAL PARAMETER INITIALIZATION:
        C
        C      SET ALL PUSHBUTTON BIT PATTERNS
0013           ICALPB="1
0014           IFVCPB="100
0015           IRSTPB="2
0016           IENDT="40
        C
        C--FOLLOWING SUBROUINE DATA INPUTS SUBJECT DATA AND CERTAIN OTHER INFO.
        C
0017           CALL DATA(IN)
        C      SWITCH ANALOG INTERFACE TO APPROPRIATE TEST TYPE
0018           TESTYP=1
0019           IF(IDATA(12).EQ.2)TESTYP=0
0021           IF(IDATA(12).EQ.3)TESTYP=2
0023           TESTYP=256*TESTYP
0024           CALL IPOKE(DROUTB,TESTYP)
0025           IF(IDATA(43).GT.0)GO TO 1000
        C      TURN ON CAL LIGHT
0027           CALL LIGHT("41)
0028    200    IRES=IPEEK(DRINB)
0029           IF(IENDT.EQ.IRES)GO TO 2000
0031           IF(ICALPB .NE. IRES) GO TO 200
0033    300    CALL CAL
        D      TYPE 1
        D1     FORMAT(' AFTER CAL')
0034           IDATA(43)=IDATA(43)+1
0035           CALL DATA(OUT)
0036    1000   CONTINUE
```

```
      D     TYPE 2
      D2    FORMAT(' AFTER 1000')
      C
      C--TURN ON APPROPIATE LIGHTS.
      C
0037        CALL LIGHT("143)
0038        IRES=0
0039        IRES=IPEEK(DRINB)
0040        IF(IRES.EQ.IENDT)GO TO 2000
      C
      C--CHECK WHICH ONE IF SO AND DO WHAT IS REQUESTED:
      C
0042        IF(ICALPB .EQ.IRES) GO TC 300   !REPEAT CALIBRATION
0044        IF(IFVCPB .EQ.IRES) CALL PFT         ! FORCED VITAL CAPACITY TEST
0046        IF(IRSTPB .EQ.IRES)CALL EX  !EXERCISE TEST
      D     TYPE 3
      D3    FORMAT(' AFTER EX')
0048        IF(IRSTPB .NE. IRES) GO TC 1000
0050  2000  CONTINUE
      D     TYPE 4
      D4    FORMAT(' AFTER 2000')
0051        CALL IPOKE(DROUTB,0)
0052        CALL DATA(OUT)
0053        STOP
0054        END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

DROUTB   000024  INTEGER*2 VARIABLE
DRINB    000026  INTEGER*2 VARIABLE
TESTYP   000030  INTEGER*2 VARIABLE
IN       000006  INTEGER*4 VARIABLE
OUT      000012  INTEGER*4 VARIABLE
ICALPB   000032  INTEGER*2 VARIABLE
IFVCPB   000034  INTEGER*2 VARIABLE
IRSTPB   000036  INTEGER*2 VARIABLE
IENDT    000040  INTEGER*2 VARIABLE
DATA     000000  REAL*4    PROCEDURE
IPOKE    000000  INTEGER*2 PROCEDURE
LIGHT    000000  INTEGER*2 PROCEDURE
IRES     000042  INTEGER*2 VARIABLE
IPEEK    000000  INTEGER*2 PROCEDURE
CAL      000000  REAL*4    PROCEDURE
PFT      000000  REAL*4    PROCEDURE
EX       000000  REAL*4    PROCEDURE


COMMON BLOCK /PDATA/     LENGTH 006000

IDATA    000000  INTEGER*2 ARRAY (1536)
MAJREV   000030  INTEGER*2 VARIABLE
MINREV   000032  INTEGER*2 VARIABLE

COMMON BLOCK /X/         LENGTH 000060

MD       000000  INTEGER*2 ARRAY (24)
```

```
      C--  SUBROUTINE INITI (VERSION 2.0),  USED TO SET THE TIMERS, COUNTERS,
      C--  & FLAGS USED IN THE ADISR INTERRUPT ROUTINE.
      C--
             SUBROUTINE INITI
             INTEGF4 BPCNT,BPCNT1,BPTIME
             INTEGER CWLCNT,AHRCNT,AHR,CWL,CSPD,CSBP,CDBP
             INTEGER AHR1,CWL1,CSPD1,CSBP1,CDBP1
             INTEGER COUNT,CNT10,CNT30,SECS,SECS30,GASDSP
             INTEGER FLRCNT,FLRFLG,EXHCNT,EXHALE
             COMMON /ADCOM1/IDUM1(34)
             COMMON/TIMCNT/COUNT,CNT10,CNT30,FLRCNT,FLRFLG,EXHCNT
             EQUIVALENCE (IDUM1(1),GASDSP),(IDUM1(10),AHRCNT)
             EQUIVALENCE (IDUM1(2),EXHALE)
             EQUIVALENCE (IDUM1(14),AHR),(IDUM1(15),AHR1)
             EQUIVALENCE (IDUM1(16),CWL),(IDUM1(17),CWL1)
             EQUIVALENCE (IDUM1(18),CWLCNT),(IDUM1(28),SECS)
             EQUIVALENCE (IDUM1(19),BPTIME),(IDUM1(22),BPCNT1)
             EQUIVALENCE (IDUM1(20),CSPD),(IDUM1(21),CSPD1)
             EQUIVALENCE (IDUM1(23),CSBP),(IDUM1(24),CSBP1)
             EQUIVALENCE (IDUM1(25),CDBP),(IDUM1(26),CDBP1)
             EQUIVALENCE (IDUM1(27),BPCNT)
             EQUIVALENCE (IDUM1(29),SECS30),(IDUM1(32),AWL)
             EQUIVALENCE (IDUM1(33),AWL1),(IDUM1(34),AWLCNT)
             EXHALE=0
             BPTIME=550
             BPCNT1=0
             BPCNT=10
             AHR=0
             AHR1=0
             CWL=0
             CWL1=0
             CSPD=0
             CSPD1=0
             CSBP=0
             CSBP1=0
             CDBP=0
             CDBP1=0
             COUNT=10
             CNT10=10
             CNT30=30
             EXHCNT=0
             CWLCNT=0
             AHRCNT=0
             SECS=0
             SECS30=0
             GASDSP=0
             AWL=0
             AWL1=0
             AWLCNT=0
             RETURN
             END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

COMMON BLOCK /ADCOM1/   LENGTH 000106

IDUM1    000000  INTEGER*2 ARRAY (34)
GASDSP   000000  INTEGER*2 VARIABLE
AHRCNT   000022  INTEGER*2 VARIABLE
EXHALE   000002  INTEGER*2 VARIABLE
AHR      000032  INTEGER*2 VARIABLE
AHR1     000034  INTEGER*2 VARIABLE
CWL      000036  INTEGER*2 VARIABLE
CWL1     000040  INTEGER*2 VARIABLE
CWLCNT   000042  INTEGER*2 VARIABLE
SECS     000066  INTEGER*2 VARIABLE
BPTIME   000044  INTEGER*2 VARIABLE
BPCNT1   000052  INTEGER*2 VARIABLE
CSPD     000046  INTEGER*2 VARIABLE
CSPD1    000050  INTEGER*2 VARIABLE
CSBP     000054  INTEGER*2 VARIABLE
CSBP1    000056  INTEGER*2 VARIABLE
CDBP     000060  INTEGER*2 VARIABLE
CDBP1    000062  INTEGER*2 VARIABLE
BPCNT    000064  INTEGER*2 VARIABLE
SECS30   000070  INTEGER*2 VARIABLE
AWL      000076  REAL*4    VARIABLE
AWL1     000100  REAL*4    VARIABLE
AWLCNT   000102  REAL*4    VARIABLE

COMMON BLOCK /TIMCNT/   LENGTH 000014

COUNT    000000  INTEGER*2 VARIABLE
CNT10    000002  INTEGER*2 VARIABLE
CNT30    000004  INTEGER*2 VARIABLE
FLRCNT   000006  INTEGER*2 VARIABLE
FLRFLG   000010  INTEGER*2 VARIABLE
EXHCNT   000012  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:04:57                 PAGE 001

0001          SUBROUTINE DATA(IOSTAT)
0002          LOGICAL*1 IFILE(12)
0003          INTEGER FILE(4)
      C
0004          INTEGER*4 IOSTAT,OUT
      C
      C
0005          COMMON /PDATA/ IPATNT(1536) ! THIS IS PATIENT DATA.
      C
      C
0006          DATA OUT/'OUT '/
0007          DATA IFILE/'D','X','1',' ',' ',' ',' ',' ',' ','D','A','T'/
      C
0008          IF(IQSET(2).NE.0) STOP 'QUEUE ALLOCATION FALIURE'
0010          IF(IOSTAT.EQ.OUT)GO TO 10
0012          TYPE 1025
0013  1025    FORMAT('$  PLEASE TYPE PATIENT DATA FILE NAME (UP TO 6 SYMBOLS):'
0014          ACCEPT 1030,(IFILE(I),I=4,9)
0015  1030    FORMAT(6A1)
      C
      C
      C
0016  10      CONTINUE
      C
      C--CONVERT ASCII FILE NAME TO RADIX 50
      C
0017          IF(IOSTAT.NE.OUT) CALL IRAD50(12,IFILE,FILE)
      C
      C--OPEN FILE:
      C
0019          ICHAN =IGETC()
0020          IF (ICHAN .LT. 0) STOP 'NO AVAILABLE CHANNEL'
0022          IF (LOOKUP(ICHAN,FILE) .LT. 0) STOP 'FILE ALLOCATION FAILURE'
      C
      C--NOW READS OR WRITES  DATA:
      C
0024          IF(IOSTAT.EQ.OUT) GO TO 1050
0026          IF(IREADW(1536,IPATNT,0,ICHAN).LT.0) STOP 'DISK READ ERROR'
0028          GO TO 1060
0029  1050    CONTINUE
0030          IF(IWRITW(1536,IPATNT,0,ICHAN).LT.0) STOP 'DISK WRITE ERROR'
0032  1060    CONTINUE
0033          CALL CLOSEC(ICHAN)
0034          RETURN
0035          END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

IFILE    000016  LOGICAL*1 ARRAY (12)
FILE     000032  INTEGER*2 ARRAY (4)
IOSTAT   000014  INTEGER*4 PARAMETER VARIABLE
OUT      000042  INTEGER*4 VARIABLE
IOSET    000000  INTEGER*2 PROCEDURE
I        000330  INTEGER*2 VARIABLE
IRAD50   000000  INTEGER*2 PROCEDURE
ICHAN    000332  INTEGER*2 VARIABLE
IGETC    000000  INTEGER*2 PROCEDURE
LOOKUP   000000  INTEGER*2 PROCEDURE
IREADW   000000  INTEGER*2 PROCEDURE
IWRITW   000000  INTEGER*2 PROCEDURE
CLOSEC   000000  REAL*4    PROCEDURE


COMMON BLOCK /PDATA/    LENGTH 006000

IPAINT   000000  INTEGER*2 ARRAY (1536)
```

```
      C--   SUBROUTINE CALIBRATION, VERSION 2.0
      C--   WRITTEN BY: ROY A. REED
      C--  DATE OCT 13, 1978
      C
      C     REWRITTEN BY: WILLIAM G. CROSIER
      C     DATE: AUG 24, 1979
      C     MODIFIED BY: ABHIJIT GADGIL
      C     DATE: DEC,1979
      C--
0001        SUBROUTINE CAL
      C--
      C--
      C--  SET UP VARIABLES
      C--
0002        LOGICAL*1 IANS,YES,NO
0003        INTEGER SEX,SSNO(3),UNIGNO(2),RETNO
0004        INTEGER CPID,DATE(3),NAME(14)
0005        INTEGER GASBUF,ADCTBF,IDATA
0006        INTEGER GASDSP,EXHALE,SVTHRS,FVC,PFT
0007        INTEGER HRTHRS,CWLCNT,AHRCNT
0008        INTEGER BPTIME,BPCNT,BPCNT1
0009        INTEGER LFLSHR,FILE(4)
0010        INTEGER PCSR,PCCB,PCS,PCC
0011        INTEGER DROUTB,DRINB,DRSR
0012        INTEGER*4 CSPD,AHR,CWL,CSBP,CDBP
0013        REAL SLPS(9),SLPST(2),SEP
0014        REAL AMBTMP,AMBPRS
      C     REAL AO2,AN2,ACO2
      C     REAL HCSV,LCSV,HCOX,LCOX,HCN2,LCN2,HCCO2,LCCO2
      C     REAL HCHR,LCHR,HCWL,LCWL,HCAS,LCAS,HCSBP,LCSBP,HCDBP,LCDBP
0015        REAL RDATA(24),CALBUF(22)
      C     REAL LSVOL,LO2,LN2,LCO2
      C     REAL HSVOL,HO2,HN2,HCO2
      C     REAL LHR,LWL,LAS,LSBP,LDBP
      C     REAL HHR,HWL,HAS,HSBP,HDBP
0016        REAL REFVAL(2,9),CALDAT(2,9),TEMP,CNT
0017        REAL*8 LHA,LOW,HIGH,AMB,CHAN(9)
      C--
      C--  SET UP COMMON AREAS
      C--
0018        COMMON /ADCOM/JCHAN(9)
0019        COMMON /PDATA/IDATA(1536)
0020        COMMON /GASCOM/GASBUF(2000)
0021        COMMON /ADCOM1/ADCTBF(34)
0022        COMMON/X/MD(24)
      C--
      C--  SET UP EQUIVALENCE STATEMENTS
      C--
0023        EQUIVALENCE (ADCTBF(1),GASDSP),(ADCTBF(2),EXHALE)
           %,(ADCTBF(3),SVTHRS),(ADCTBF(4),FVC)
           %,(ADCTBF(8),PFT),(ADCTBF(10),AHRCNT),(ADCTBF(11),HRTHRS)
           %,(ADCTBF(14),AHR)
           %,(ADCTBF(16),CWL)
           %,(ADCTBF(18),CWLCNT),(ADCTBF(19),BPTIME)
```

```
                  %,(ADCTBF(20),CSPD)
                  %,(ADCTBF(22),BPCNT1),(ADCTBF(23),CSBP)
                  %,(ADCTBF(25),CDBP)
                  %,(ADCTBF(27),BPCNT)
                  %,(ADCTBF(30),LFLSHR)
0024               EQUIVALENCE (IDATA(2),SEX),(IDATA(3),SSNO(1))
                  %,(IDATA(7),UNIGNO(1)),(IDATA(9),RETNO)
                  %,(IDATA(10),CPID)
                  %,(IDATA(26),NAME(1))
                  %,(IDATA(40),DATE(1))
0025               EQUIVALENCE (IDATA(73),RDATA(1))
0026               EQUIVALENCE (IDATA(54),AMBTMP),(IDATA(57),AMBPRS)
            C      EQUIVALENCE(CALBUF(1),HCSV),(CALBUF(2),LCSV)
            C      %,(CALBUF(3),HCOX),(CALBUF(4),LCOX)
            C      %,(CALBUF(5),HCN2),(CALBUF(6),LCN2)
            C      %,(CALBUF(7),HCCO2),(CALBUF(8),LCCO2)
            C      %,(CALBUF(9),HCHR),(CALBUF(10),LCHR)
            C      %,(CALBUF(19),HCSBP),(CALBUF(20),LCSBP)
            C      %,(CALBUF(21),HCDBP),(CALBUF(22),LCDBP)
0027               EQUIVALENCE (CALBUF,REFVAL)
            C--
            C--   SET UP DATA VALUES
            C--
0028               DATA SLPS/.0025,.0300,.0245,.0025,.0720,.0757,.0249,.0623,.0624/
0029               DATA SLPST/0.01,0.0025/
0030               DATA SEP/.02/
0031               DATA YES/1HY/,NO/1HN/
0032               DATA DROUTB/"167772/,DRINB/"167774/,DRSR/"167770/
0033               DATA PCSR/"172540/,PCCB/"172542/
0034               DATA PCS/"113/,PCC/"144/
            C      DATA MD/'JA','N ','FE','B ','MA','R ','AP','R ',
            C      @    'MA','Y ','JU','N ','JU','L ','AU','G ','SE','P ',
            C      @    'OC','T ','NO','V ','DE','C '/
0035               DATA LOW,HIGH,AMB/'  LOW   ','  HIGH   ','AMBIENT '/
0036               DATA CHAN/'SP. VOL.',' OXYGEN ','NITROGEN','   CO2  ',
                  @  'HT.RATE','WL/ELEV.','  SPEED ',' SYS. BP','DIAS. BP'/
0037               IF(IDATA(12).EQ.2) SLPS(6)=SLPST(1)
0039               IF(IDATA(12).EQ.2) SLPS(7)=SLPST(2)
            C--
            C--   READ IN THE CALIBRATION FILE, WHICH CONTAINES THE STANDARD DATA
            C--   USED IN CALIBRATION OF SLOPES AND INTERCEPTS.
            C--
0041               CALL IRAD50(12,'DXOCALCONCAT',FILE)
0042               ICHAN=IGETC()                                    !GET A I/O CHANNEL
0043               IF(ICHAN.LT.0)STOP 'CHANNEL ALLOCATION FAILURE IN CAL'
0045               IF(LOOKUP(ICHAN,FILE).LT.0)STOP 'FILE ALLOCATION FAILURE IN CAL'
0047               IF(IREADW(44,CALBUF,1,ICHAN).LT.0) STOP 'DISK READ ERROR IN CAL'
0049               IF(IDATA(12).EQ.2)GO TO 5
0051               REFVAL(1,7)=CALBUF(15)                     !BIKE SPEED      HIGH REF. VAL
0052               REFVAL(2,7)=CALBUF(16)                     !                LOW REF. VALUE
0053               GO TO 6
0054        5      CONTINUE
0055               REFVAL(1,6)=CALBUF(13)                     !TREADMILL ELEVATION    HIGH
0056               REFVAL(2,6)=CALBUF(14)                     !                       LOW
```

```
0057          REFVAL(1,7)=CALBUF(17)              !TREADMILL SPEED          HIGH
0058          REFVAL(2,7)=CALBUF(18)              !                         LOW
0059    6     CONTINUE
0060          REFVAL(1,8)=CALBUF(19)              !SBP                      HIGH
0061          REFVAL(2,8)=CALBUF(20)              !                         LOW
0062          REFVAL(1,9)=CALBUF(21)              !DBP                      HIGH
0063          REFVAL(2,9)=CALBUF(22)              !                         LOW
0064    7     CONTINUE
0065          IF(IDATA(11).EQ.1) GO TO 10
        C-    INITTIALIZE PROGRAMMABLE GAIN CODE BITS IN A/D COMMAND
        C-    WORD ARRAY (LATER USED BY INTERRUPT SERVICE ROUTINE)
0067          IAD="10
0068          DO 9 ICH=1,9
0069          JCHAN(ICH)=IAD
0070    9     IAD=IAD+"400
0071    10    CONTINUE
        C--
        C     INITIALIZE PARAMETERS
        C--
0072          PFT=0
0073          FVC=0
0074          HRTHRS=-2048
0075          SVTHRS=-2048
0076          LFLSHR="1                 !SET CAL LIGHT TO FLASH
        C
        C     COLLECT LOW CAL DATA
        C
0077          MODE=1                           !         SET LOW CAL MODE
0078          ISUB=2                           !         USE SECOND ROW OF CALDAT ARR
0079          DELAY=5.0             !          SET DELAY AT 5 SECONDS
0080          LHA=LOW                          !         SET ALPHA IDENTIFIER FOR CAL
0081          ASSIGN 162 TO IRETRN             !         ASSIGN RETURN LABEL
0082          GO TO 5000            !          GO TO CAL DATA COLLECTION/AVERAGING
        C
        C     COLLECT HIGH CAL DATA
0083    162   MODE=2                           !         SET HIGH CAL MODE
0084          ISUB=1                           !         USE FIRST ROW OF CALDAT ARRA
0085          DELAY=5.0             !          SET DELAY AT 5 SECONDS
0086          LHA=HIGH              !          SET ALPHA IENTIFIER FOR CAL MODE
0087          ASSIGN 262 TO IRETRN             !         ASSIGN RETURN LABEL
0088          GO TO 5000            !          GO TO DATA COLLECTION/AVERAGING ROUTI
        C--
        C-- CALCULATE THE SLOPES AND Y INTERCEPTS FOR ALL 9 CHANNELS
0089    262   K=7
0090          DO 265 ICHAN=1,9
0091          RDATA(K)=0.0
0092          DENOM=CALDAT(1,ICHAN)-CALDAT(2,ICHAN)
0093          IF(DENOM.NE.0.0)RDATA(K)=(REFVAL(1,ICHAN)-REFVAL(2,ICHAN))/DENOM
0095          K=K+1
0096          RDATA(K)=REFVAL(2,ICHAN)-RDATA(K-1)*CALDAT(2,ICHAN) !Y-INT.
0097    265   K=K+1
        C
        C     COLLECT AMBIENT AIR DATA
        C
```

```
              MODE=3                          !       SET AMBIENT AIR MODE
              ISUB=1                          !       USE FIRST ROW OF CALDAT ARRAY
              DELAY=20.0          !       SET DELAY 20 SECONDS
              LHA=AMB                         !       SET ALPHA IENTIFIER FOR CAL MODE
              ASSIGN 370 TO IRETRN            !       ASSIGN RETURN LABEL
              GO TO 5000          !       GO TO DATA COLLECTION/AVERAGING ROUTINE
        C--
        C-- PH20=VAPOR PRESSURE OF WATER AT AMBIENT
        C--       PRESSURE AND TEMPERATURE.
        C--
        370   PH20=.0369*AMBTMP*AMBTMP-.4012*AMBTMP+10.76
        C--
        C--  STPD=FACTOR TO ADJUST VOLUME AT AMBIENT CONDITIONS
        C--   TO VOLUME AT STANDARD TEMPERATURE AND PRESSURE (DRY).
        C--   NORMAL BODY TEMPERATURE IS 310 DEGREES KELVIN
        C--
              BTPS=310.0/(AMBTMP+273.0)
              STPD=(273.16/(AMBTMP+273.16))*((AMBPRS-PH20)/760.)
        C--
        C--  CALCULATE THE PERCENTS
        C--
              RDATA(3)=CALDAT(1,2)*RDATA(9)+RDATA(10)              !AMBIENT O2
              RDATA(4)=CALDAT(1,3)*RDATA(11)+RDATA(12)     !AMBIENT N2
              RDATA(5)=CALDAT(1,4)*RDATA(13)+RDATA(14)     !AMBIENT CO2
              RDATA(1)=STPD                                !STPD FACTOR
              RDATA(2)=BTPS                                !BTPS FACTOR
        C--
        C--
        C--  SWITCH ANALOG INTERFACE TO NORMAL MODE
        C--
              CALL IPOKE(DRSR,0)
        C--
        C--  WRITE REPORT
        C--
              IMM=2*DATE(1)
              IMM1=IMM-1
        999   TYPE 1000
        1000  FORMAT(//32X,'CALIBRATION REPORT',/36X,'CDAS - V2.0')
              TYPE 1010,CPID,(UNIGNO(I),I=1,2),RETNO,DATE(2),MO(IMM1),
             @MO(IMM),DATE(3)
              TYPE 1020,(NAME(I),I=1,14),SEX,(SSNO(I),I=1,3)
        1010  FORMAT(//,5X,'CPID NO.:  ',I5,28X,'UNIQUE NO.: 'A2,I4,//,
             %5X,'RETEST NO.:   ',I2,28X,'DATE:       ',I2,1X,2A2,I2,//)
        1020  FORMAT(5X,'SUBJECT NAME:  '14A2,1X,'SEX:         ',A2,//,
             %5X,'SOC. SEC. NO.: ',I3,'-',I2,'-',I4)
              TYPE 1030
        1030  FORMAT(//,5X,'AMBIENT CONDITIONS:',23X,'CALIBRATION FACTORS:'
             %,//,40X,'CHANNEL        SLOPE        Y-INT.')
              TYPE 1040,AMBTMP,RDATA(7),RDATA(8)
             %,AMBPRS,RDATA(9),RDATA(10)
             %,(RDATA((I-9)/2),RDATA(I),RDATA(I+1),I=11,19,2)
             %,(RDATA(I),I=21,24)
        1040  FORMAT(/,5X,'TEMP.:         ',F4.1,' C',16X,'SP.VOL',5X,F9.5,3X,F8.3,
             %/,5X,'PRES.:       ',F5.1,' MM HG',12X,'O2           ',F9.5,3X,
              %F8.3,/,5X,'STPD FACTOR:  ',F5.3,16X,'N2           ',F9.5,3X,F8.3
              %/,5X,'BTPS FACTOR:  ',F5.3,16X,'CO2         ',F9.5,3X,F8.3
              %/,5X,'O2:              ',F4.1,'%',17X,'HR          ',F9.5,3X,F8.3,
              %/,5X,'N2:              ',F4.1,'%',17X,'ELEV/WL     ',F9.5,3X,F8.3,
              %/,5X,'CO2:             ',F4.2,'%',17X,'SPEED       ',F9.5,3X,F8.3,
              %/,40X,'SBP          ',F9.5,3X,F8.3,
              %/,40X,'DBP          ',F9.5,3X,F8.3)
0125           IRA=1
0126           IF(IDATA(11).EQ.2)IRA=1.414
0128           LINE=29
0129           DO 3000 IC=1,9
0130           IS=(IC-1)*2+7
0131           SMAX=SLPS(IC)*IRA+(SLPS(IC)*IRA*SEP)
0132           SMIN=SLPS(IC)*IRA-(SLPS(IC)*IRA*SEP)
0133           IF(RDATA(IS).LT.SMAX.AND.RDATA(IS).GT.SMIN) GO TO 3000
0135           PERER=((RDATA(IS)-SLPS(IC))/SLPS(IC))*100.0
0136           LINE=LINE+2
0137           TYPE2010, CHAN(IC),PERER
0138    2010   FORMAT(/1X,A8,' CALIB. OUT OF RANGE BY ',F6.1, '%')
0139    3000   CONTINUE
0140           LINE=66-LINE
0141           DO 3100 IL=1,LINE
0142    3100   TYPE 3150
0143    3150   FORMAT(1X)
        C
        C      SET TIME OF DAY TIMER IF SYSTEM CLOCK SHOWS LESS THAN
        C     5 HOURS (18000 SECS) SINCE MIDNIGHT OR SYSTEM BOOT, OR
        C      IF WE ARE RUNNING PLAYBACK MODE (SINCE SYSTEM CLOCK TIME
        C      WILL NOT BE TEST TIME).
        C
0144           IF (SECNDS(0.).LT.18000..OR.IDATA(11).EQ.2) CALL TIMSET
        C--
        C--   SET LIGHTS ON FRONT PANNEL
        C--
0146           CALL LIGHT("1)
        C--
        C--   ALL DONE RETURN TO PROCES
        C--
0147           RETURN
        C
        C      ROUTINE TO COLLECT & AVERAGE DATA FROM A/D CONVERTER
        C      FOR CALIBRATION
        C
        C      INITIALIZE COUNTERS, TIMERS, & FLAGS
0148    5000   CALL INITI
0149           BPTIME=0
0150           BPCNT=40
0151           BPCNT1=40
        C      IS TEST BEING RUN IN REAL TIME(LIVE) ?
0152           IF (IDATA(11).EQ.1) GO TO 5010
0154    5006   TYPE 5007, LHA
0155    5007   FORMAT (/' IS TAPE READY FOR ',A7,'CALIB. (Y OR N) ?',$)
0156           ACCEPT 5008,IANS
0157    5008   FORMAT(A1)
```

```
0158          IF (IANS.NE.YES) GO TO 5006
0160    5010  CONTINUE
0161          CALL LIGHT ("1)                 ! TURN ON CAL LIGHT
0162          CALL IPOKE(DRSR,MODE)           ! SEND CAL SIGNAL TO LAB EQUIP.
0163          TI=SECNDS(0.0)
0164    5015  DELTA=SECNDS(TI)
0165          IF (DELTA.LT.CELAY) GO TO 5015 ! WAIT FOR SIGNALS TO STABILIZE
0167          CALL IPOKE (PCCB,PCC)                    ! SET PROGRAMMAPLE CLOCK FOR 100
0168          CALL IPOKE (PCSR,PCS)                    ! START CLOCK FOP DATA ACQUISITI
0169    5020  IF (BPCNT1.GT.0) GO TO 5020 ! WAIT UNTIL 4 SEC. OF DATA IS COLLECTEC
0171          CALL IPOKE (PCSR,0)             ! STOP CLOCK
0172          CALL LIGHT ("1)                          ! TURN ON CAL LIGHT
        C     DO 5051 I=1, GASDSP,8
        C     TYPE 5050, I, (GASBUF(J),J=I,I+7)  ! TYPE GAS BUFFER CONTENTS
0173    5050  FORMAT (/1X,I4,')',8(1X,I6))
0174    5051  CONTINUE
        C
        C     CALCULATE AVG. VALUE FOR EACH OF THE 9 CHANNELS
        C
0175          DO 5080 K=1,9
0176    5080  CALDAT(ISUB,K)=0.0              ! INITIALIZE VALUES FOR SUMMING
0177          SAMP=FLOAT(GASDSP)/4.0
0178          DO 6010 K=1,4
0179          DO 6000 IAV=1, GASDSP,4                  ! CHAN. 1-4 ARE GAS CHANNELS
0180    6000  CALDAT(ISUB,K)=CALCAT(ISUB,K) + FLOAT(GASPUF(IAV+K-1))
0181    6010  CALDAT(ISUB,K)=CALDAT(ISUB,K)/SAMP
        D     TYPE 6050,LHA,GASDSP,SAMP,(CALDAT(ISUB,K),K=1,4)
0182    6050  FORMAT(/1X,A8,' CAL VALUES'/'  GASDSP     SAMP      SP VOL',6X,'02',
       @  8X,'N2',7X,'CC2'/I10,5F10.2)
        C
        C     CALCULATE AVERAGES FOR CHAN. 5-9
        C
0183          IF(IAJFLT(AHR,TEMP).EQ.-2) TYPE 9999
0185          IF(AHRCNT.NE.0) CALDAT(ISUB,5)=TEMP/FLOAT(AHRCNT)
0187          IF(IAJFLT(CWL,TEMP).EQ.-2) TYPE 9999
0189          IF(CWLCNT.NE.0) CALDAT(ISUB,6)=TEMP/FLOAT(CWLCNT)
0191          IF(IAJFLT(CSPD,TEMP).EQ.-2) TYPE 9999
0193          IF(CWLCNT.NE.0) CALDAT (ISUB,7)=TEMP/FLOAT(CWLCNT)
0195          CNT=FLOAT(BPCNT-BPCNT1)
0196          IF(CNT.EQ.0.0) GO TO 6055
0198          IF(IAJFLT(CSBP,TEMP).EQ.-2) TYPE 9999
0200          CALDAT(ISUB,8)=TEMP/CNT
0201          IF(IAJFLT(CDBP,TEMP).EQ.-2) TYPE 9999
0203          CALDAT(ISUB,9)=TEMP/CNT
0204    6055  CONTINUE
        D     TYPE 6060,AHRCNT,CWLCNT,BPCNT,BPCNT1,CNT,(CALDAT(ISUB,K),K=5,9)
0205    6060  FORMAT (/'  AHRCNT CWLCNT BPCNT BPCNT1    CNT'/4I7,F7.2/
       @  5X,'HR',8X,'WL',6X,'SPEED',6X,'SBP',7X,'DBP'/5F10.2)
        C
0206          GO TO IRETRN                    ! RETURN
        C
0207    9999  FORMAT(/,1X,'OVERFLOW IN INTEGER*4 TO REAL CONVERSION')
0208          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

FILE    000014  INTEGER*2 ARRAY (4)
SLPS    000024  REAL*4    ARRAY (9)
SLPST   000070  REAL*4    ARRAY (2)
CALBUF  000100  REAL*4    ARRAY (22)
REFVAL  000100  REAL*4    ARRAY (2,9) VECTORED
CALDAT  000230  REAL*4    ARRAY (2,9) VECTORED
CHAN    000340  REAL*8    ARRAY (9)
IANS    002676  LOGICAL*1 VARIABLE
YES     000454  LOGICAL*1 VARIABLE
NO      000455  LOGICAL*1 VARIABLE
PCSR    000464  INTEGER*2 VARIABLE
PCCB    000466  INTEGER*2 VARIABLE
PCS     000470  INTEGER*2 VARIABLE
PCC     000472  INTEGER*2 VARIABLE
DROUTB  000456  INTEGER*2 VARIABLE
DRINB   000460  INTEGER*2 VARIABLE
DRSR    000462  INTEGER*2 VARIABLE
SEP     000450  REAL*4    VARIABLE
TEMP    002700  REAL*4    VARIABLE
CNT     002704  REAL*4    VARIABLE
LHA     002710  REAL*8    VARIABLE
LOW     000474  REAL*8    VARIABLE
HIGH    000504  REAL*8    VARIABLE
AMB     000514  REAL*8    VARIABLE
IRAD50  000000  INTEGER*2 PROCEDURE
ICHAN   002720  INTEGER*2 VARIABLE
IGETC   000000  INTEGER*2 PROCEDURE
LOOKUP  000000  INTEGER*2 PROCEDURE
IREADW  000000  INTEGER*2 PROCEDURE
IAD     002722  INTEGER*2 VARIABLE
ICH     002724  INTEGER*2 VARIABLE
MODE    002726  INTEGER*2 VARIABLE
ISUB    002730  INTEGER*2 VARIABLE
DELAY   002732  REAL*4    VARIABLE
IRETRN  002736  INTEGER*2 VARIABLE
K       002740  INTEGER*2 VARIABLE
DENOM   002742  REAL*4    VARIABLE
PH2O    002746  REAL*4    VARIABLE
BTPS    002752  REAL*4    VARIABLE
STPD    002756  REAL*4    VARIABLE
IPOKE   000000  INTEGER*2 PROCEDURE
IMM     002762  INTEGER*2 VARIABLE
IMM1    002764  INTEGER*2 VARIABLE
I       002766  INTEGER*2 VARIABLE
IRA     002770  INTEGER*2 VARIABLE
LINE    002772  INTEGER*2 VARIABLE
IC      002774  INTEGER*2 VARIABLE
IS      002776  INTEGER*2 VARIABLE
SMAX    003000  REAL*4    VARIABLE
SMIN    003004  REAL*4    VARIABLE
PERER   003010  REAL*4    VARIABLE
IL      003014  INTEGER*2 VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

SECNDS   000000  REAL*4      PROCEDURE
TIMSET   000000  REAL*4      PROCEDURE
LIGHT    000000  INTEGER*2   PROCEDURE
INITI    000000  INTEGER*2   PROCEDURE
TI       003016  REAL*4      VARIABLE
DELTA    003022  REAL*4      VARIABLE
SAMP     003026  REAL*4      VARIABLE
FLOAT    000000  REAL*4      PROCEDURE
IAV      003032  INTEGER*2   VARIABLE
IAJFLT   000000  INTEGER*2   PROCEDURE


COMMON BLOCK /ADCOM/     LENGTH 000022

JCHAN    000000  INTEGER*2 ARRAY (9)

COMMON BLOCK /PDATA/     LENGTH 006000

IDATA    000000  INTEGER*2 ARRAY (1536)
SEX      000002  INTEGER*2 VARIABLE
SSNO     000004  INTEGER*2 ARRAY (3)
UNIQNO   000014  INTEGER*2 ARRAY (2)
RETNO    000020  INTEGER*2 VARIABLE
CPID     000022  INTEGER*2 VARIABLE
NAME     000062  INTEGER*2 ARRAY (14)
DATE     000116  INTEGER*2 ARRAY (3)
RDATA    000220  REAL*4    ARRAY (24)
AMBTMP   000152  REAL*4    VARIABLE
AMBPRS   000160  REAL*4    VARIABLE

COMMON BLOCK /GASCOM/    LENGTH 007640

GASBUF   000000  INTEGER*2 ARRAY (2000)

COMMON BLOCK /ADCOM1/    LENGTH 000104

ADCTBF   000000  INTEGER*2 ARRAY (34)
GASDSP   000000  INTEGER*2 VARIABLE
EXHALE   000002  INTEGER*2 VARIABLE
SVTHRS   000004  INTEGER*2 VARIABLE
FVC      000006  INTEGER*2 VARIABLE
PFT      000016  INTEGER*2 VARIABLE
AHRCNT   000022  INTEGER*2 VARIABLE
HRTHRS   000024  INTEGER*2 VARIABLE
AHR      000032  INTEGER*4 VARIABLE
CWL      000036  INTEGER*4 VARIABLE
CWLCNT   000042  INTEGER*2 VARIABLE
BPTIME   000044  INTEGER*2 VARIABLE
CSPD     000046  INTEGER*4 VARIABLE
BPCNT1   000052  INTEGER*2 VARIABLE
CSBP     000054  INTEGER*4 VARIABLE
CDBP     000060  INTEGER*4 VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

BPCNT    000064  INTEGER*2 VARIABLE
LFLSHR   000072  INTEGER*2 VARIABLE

COMMON BLOCK /X/          LENGTH 000060

MD       000000  INTEGER*2 ARRAY (24)
```

```
      C--  FORCED VITAL CAPACITY SUBROUTINE, VERSION 2.0
      C--
0001          SUBROUTINE PFT
      C--
      C--  SET UP VARIABLES
      C--
0002          INTEGER AGE
      C--  !PATIENTS AGE AND HEIGHT, TO BE USED IN
      C--  !THE CALCULATIONS OF THE PREDICTD VALUES.
0003          INTEGER CPID
      C--  !PATIENTS CARDIOPULMONARY IDENTITY NUMBER.
0004          INTEGER NAME(14)
      C--  !PATIENT'S NAME.
0005          INTEGER SSNO(3)
      C--  !PATIENT'S SOCIAL SECURITY NUMBER.
0006          INTEGER DATE(3)
      C--  !DATE THE TEST WAS  CONDUCTED.
0007          INTEGER RETNO
      C--  !PATIENT'S RETEST NUMBER.
0008          INTEGER UNIQNO(2)
      C--  !PATIENT'S UNIQUE NUMBER.
0009          INTEGER M,F,SEX
      C--  !SUBJECTS SEX TO BE USED IN CALCULATIONS.
0010          INTEGER ADCTBF(34)
      C--  !COMMON AREA FOR A/D COMMAND VARIABLES
0011          INTEGER GASBUF(2000)
      C--  !A/D COMMON DATA BUFFER.
0012          INTEGER GASDSP
      C--  !GAS BUFFER DISPLACEMENT
0013          INTEGER SVTHRS
      C--  !SPIROMETER VOLUMN THRESHOLD.
0014          INTEGER FVC
      C--  !FLAG (=-1FOR FORCED VITAL CAPACITY
      C    !EQUAL TO 0 FOR ALL OTHERS).
0015          INTEGER PFT
      C--  !FLAG =-1
0016          INTEGER PUSHES
      C--  !CONTAINES 1'S FOR BUTTONS THAT ARE
      C    !PUSHED.
0017          INTEGER PCSR
      C--  !ADDRESS OF THE PROGRAMABLE CLOCK
      C    !STATUS REGISTER.
0018          INTEGER PCCB
      C--  !ADDRESS OF THE PROGRAMABLE CLOCK
      C    !COUNTER BUFFER.
0019          INTEGER EXHALE
      C--  !DONE FLAG SET BY THE A/D INTERRUPT
      C    !ROUTINE.
0020          INTEGER PCS
      C--  !HAS THE VALUE TO BE PLACED INTO THE
      C    !CLOCK STATUS REGISTER.
0021          INTEGER PCC
      C--  !HAS THE VALUE TO BE PLACED INTO THE
      C    !CLOCK COUNTER BUFFER.
```

```
0022          INTEGER OLD
        C--  !VARIABLE TO HOLD THE OLD VALUE OF EXHAST.
0023          INTEGER DROUTB,DRINB
        C--  !VARIBLES THAT CONTAIN THE ADDRESS OF THE
        C--  !DR11 PARRELL INTERFACE CARD INPUT AND OUTPUT BUFFERS.
0024          REAL HIGHT,BPTS
        C--  !PATIENT'S HEIGHT IN CM.
0025          REAL SLOPE,YNTRCP
        C--  !VARIEBLES THAT CONTAIN INFORMATION PASSED
        C--  !THROUGH GENCOM COMMON BLOCK TO CALCULATE THE
        C--  !SPIROMETER THRESHOLD.
        C--
        C--  SET UP THE COMMON AREAS.
        C--
0026          COMMON/PDATA/IDATA(1536)
        C--  !SET UP THE COMMON BUFFER FOR PATIENT DATA
        C--  !ALSO, CONTAINES THE PATIENTS INFORMATION.
0027          COMMON /GASCOM/GASBUF
        C--  !DATA BUFFER COMMON AREA.
0028          COMMON /ADCOM1/ADCTBF
        C--  !COMMON AREA TO PASS COMMAND VARIABLES AND
        C    !CONSTANTS.
0029          COMMON/X/MD(24)
0030          EQUIVALENCE (IDATA(85),SLOPE)
        C--  !CONTAINES THE SLOPE FROM CALIBRATION ROUTINE.
0031          EQUIVALENCE (IDATA(87),YNTRCP)
        C--  !CONTAINES THE Y-INTERCEPT FROM THE CALIBRATION ROUTINE.
0032          EQUIVALENCE(IDATA(3),SSNO(1))
        C--  !THESE THREE DATA WORDS CONTAIN
        C--  !THE PATIENT'S SOCIAL SECURITY NUMBER.
0033          EQUIVALENCE (IDATA(7),UNIQNO(1))
        C--  !THIS DATA WORD  CONTAINES THE PATIENT'S
        C--  !UNIQUE NUMBER.
0034          EQUIVALENCE (IDATA(9),RETNO)
        C--  !THIS DATA WORD CONTAINES THE SUBJECT'S
        C--  !RETEST NUMBER.
0035          EQUIVALENCE (IDATA(10),CPID)
        C--  !THIS DATA WORD CONTAINES THE PATIENT'S
        C--  !CARDIOPULMUNARY NUMBER.
0036          EQUIVALENCE(IDATA(40),DATE(1))
        C--  !THESE THREE DATA WORDS CONTAIN THE
        C--  !PATIENT'S TEST DATE.
0037          EQUIVALENCE (IDATA(26),NAME(1))
        C--  !THE 14 DATA WORDS CONTAIN THE
        C--  !PATIENT'S NAME.
0038          EQUIVALENCE (IDATA(2),SEX)
        CC    !THIS DATA WORD CONTAINES THE PATIENTS SEX.
0039          EQUIVALENCE (IDATA(25),AGE)
        C--  !THIS DATA WORD CONTAINES THE PATIENTS AGE.
0040          EQUIVALENCE (IDATA(50),HIGHT)
        C--  !THIS DATA WORD CONTAINES THE PATIENTS HEIGHT.
0041          EQUIVALENCE (IDATA(75),BPTS)
0042          EQUIVALENCE (ADCTBF(1),GASDSP)
        C--   GAS BUFFER DISPLACEMENT
```

```
0043            EQUIVALENCE (ADCTBF(2),EXHALE)
          C--  !DONE FLAG SET BY THE A/D INTERRUPT ROUTINE.
0044            EQUIVALENCE (ADCTBF(3),SVTHRS)
          C--  !SPIROMETER VOLUMN THRESHOLD, MUST BE CALCULATED.
0045            EQUIVALENCE(ADCTBF(4),FVC)
          C--  !FLAG EQUAL TO -1 FOR FORCED VITIAL CAPACITY.
0046            EQUIVALENCE (ADCTBF(7),PUSHES)
          C--  !CONTAINES 1'S FOR BUTTONS THAT ARE
          C    !PUSHED BY PROCESS.
0047            EQUIVALENCE (ADCTBF(8),PFT)
          C--  !FLAG FOR THE PFT TEST, MUST BE SET TO -1.
0048            EQUIVALENCE (ADCTBF(30),LFLSHR)
          C--  !CONTAINS LIGHT TO FLASH
0049            EQUIVALENCE (IDATA(129),RFVC)
          C--  ! CONTAINES THE FORCED VITAL CAPACITY IN THE
          C--  !PATIENTS DATA BUFFER, WHICH WILL BE STORED
          C--  !ON THE FLOPPY DISK.
0050            EQUIVALENCE(IDATA(131),FEV1)
          C--  !CONTAINES THE FORCED EXPIRATORY VOLUMN IN 1 SEC.
0051            EQUIVALENCE (IDATA(133),PEFR)
          C--  !CONTAINES THE PEAK EXPIRATORY FLOW RATE.
0052            EQUIVALENCE (IDATA(135),FEF)
          C--  !CONTAINES THE FORCED EXPIRATORY FLOW BETWEEN 25% AND 75% MAX FVC.
0053            EQUIVALENCE (IDATA(137),FEF212)
          C--  !CONTAINES THE MEANS FORCED EXPIRATORY FLOW BETWEEN 200ML. AND 1200
0054            EQUIVALENCE (IDATA(141),F1FP)
          C--  !CONTAINES THE % FEV1/FVC.
0055            EQUIVALENCE (IDATA(145),PFVC)
          C--  !CONTAINES THE PREDICTED FORCED VITIAL CAPACITY.
0056            EQUIVALENCE (IDATA(147),OPFVCP)
          C--  !CONTAINES THE % OBSERVED FVC/ PREDICTED FVC.
0057            EQUIVALENCE (IDATA(149),PFEV1)
          C--  !CONTAINES THE PREDICTED FORCED EXPIRATORY VOLUMN IN 1 SEC.
0058            EQUIVALENCE (IDATA(151),OPFEV1)
          C--  !CONTAINES THE % OBSERVED FEV1 / PREDICTED FEV1.
          C--
          C--  INITILALIZE PARAMETERS
          C--
0059            DATA M/'M'/,F/'F'/
          C--  !SET VAIABLES M AND F EQUAL TO THEIR ACSII VALUES.
0060            DATA DROUTB/"167772/
          C--  !DATA OUT BUFFER FOR THE PARRELL INTERFACE CARD.
0061            DATA DRINB/"167770/
          C--  !DATA IN BUFFER FOR THE PARRELL INTERFACE CARD.
0062            DATA PCSR/"172540/
          C--  !OCTAL ADDRESS OF THE PROGRAMABLE CLOCK(KW11-P)
          C    !STATUS REGISTER.
0063            DATA PCCB/"172542/
          C--  !OCTAL ADDRESS OF THE PROGRAMABLE CLOCK
          C    !COUNTER BUFFER.
0064            DATA PCC/"144/
          C--  !SET CLOCK TO GET 100 INTERRUPS/SEC.
0065            DATA PCS/"113/
          C     DATA MO/'JA','N ','FE','B ','MA','R ','AP','R ',
      C     @'MA','Y ','JU','N ','JU','L ','AU','G ','SE','P ',
      C     @   'OC','T ','NO','V ','DE','C '/
      C--   !OCTAL VALUE TO BE LOADED INTO THE CLOCK STATUS
      C     !REGISTER. BITS 0,1,3, AND 6 ARE TURNED ON.
      C     !   BIT 6 - ALLOWS DONE TO CAUSE AN INTERRUPT.
      C     !   BIT 3 - SELECTS REPEATED INTERRUPT MODE.
      C     !   BIT 1 - SET CLOCK RATE TO 10KHZ.
      C     !   BIT 0 - STARTS THE CLOCK.
0066         FVC=-1
      C--   !SET A/D FOR FORCED VITAL CAPACITY DATA ACQUISITION.
0067         PFT=-1
      C--   !INITILIZE THE PFT FLAG.
0068         SVTHRS=IFIX((0.1-YNTRCP)/SLOPE)
      C--   !CALCULATE THE SPIROMETER THRESHOLD .
      C--
      C--   SET UP CLOCK STATUS AND COUNTER BUFFER.
      C--
0069         CALL IPOKE(DRCUTB,0)
      C--   !INITILIZE THE DRV11 OUT BUFFER TO 0.
0070  10     CONTINUE
0071         CALL INITI
      C--   !SET TIMERS TO INITIAL VALUES.
0072         EXHALE=0
0073         OLD=0
      C--   !SET FLAGS TO 0
0074         CALL LIGHT ("100)
0075                LFLSHR="100
      C--   !SET THE BIT TO CAUSE THE FVC LIGHT TO FLASH.
0076         CALL IPOKE(PCCB,PCC)
      C--   !SET CLOCK COUNTER FOR 100 INTERRUPTS PER. SECOND.
0077         CALL IPOKE(PCSR,PCS)
      C--   !TURN THE CLOCK ON. IPOKE PUTS THE VALUES INTO THE
      C     !SPECIFIED ADDRESS.
      C
      C--   CHECK FOR THE DONE FLAG OR USER TERMINATION
      C--
0078         TI=SECNDS(0.0)
      C--   !SET START TIME TO 0 FOR ELAPSED TIME CHECK.
0079  100    DELTA=SECNDS(TI)
      C--   !GET THE ELAPSED TIME.
0080         IF(EXHALE.EQ.0.AND.OLD.EQ.-1) GO TO 1000
      C--    !TEST FOR DONE FLAG FROM THE A/D.
      C--   !BRANCH TO RETURN ON SUCESSFULL COMPLETION.
0082         OLD=EXHALE
      C--   !SET OLD EQUAL TO THE CURRENT VALUE  OF EXHALE.
0083         IF(DELTA.LT.10.0) GO TO 100
      C--   !CHECK FOR 10 SECONDS OF ELAPSED TIME. IF NOT GREATER THAN
      C--   !10 BRANCH TO 10 TO RECHECK.
0085         TYPE 200
0086  200    FORMAT(/,1X,'FORCED VITAL CAPACITY TEST TIMED OUT',
            %/,1X,'PLEASE RETEST SUBJECT',64(/))
0087         CALL IPOKE(PCSR,0)
      C--   !TURN THE CLOCK OFF.
0088         GO TO 2200
```

```
        C--  !BRANCH TO 10, FVC TEST TIMED OUT, TRY IT AGAIN.
0089    1000  CONTINUE
0090          CALL IPOKE(PCSR,0)
        C--  !TUPN THE CLOCK OFF.
        D     TYPE 999,(GASBUF(I),I=1,GASDSP)
0091    999   FORMAT(/,1X,8(2X,I6))
0092          IF(GASDSP.GT.100)GO TO 1050
0094    1010  T"PE 1001
0095    1001  FORMAT(/,1X,'EXPIRATION BREATH TOO SHORT:',
             %/,1X,'PLEASE RETEST SUBJECT',64(/))
        D     TYPE 1002
0096    1002  FORMAT(/,'$ TYPE IN THE S. VOL. THRESHOLD ')
        D     ACCEPT 1003,SVTHRS
0097    1003  FORMAT(I6)
0098          GO TO 2200
0099    1050  CONTINUE
0100          CALL LIGHT("100)
        C--
        C--  GET THE FVC.
        C--
0101          GASAVG=0
        C--   INITILIZE THE GAS AVERGER TO 0.
0102          IPKVL=-2047
        C--  !SET PEKK VALUE TO 0
0103          DO 1100 IPK=1,GASDSP
0104          IF(GASBUF(IPK).LT.IPKVL)GO TO 1100
0106          IPKVL=GASBUF(IPK)
0107          IPEAK=IPK
0108    1100  CONTINUE
        C--  !THIS DO LOOP GET MAX PEEK VALUE
0109          IF(IPEAK.LT.100) GO TO 1010
0111          DO 1150 IAVG=IPEAK-3,IPEAK
        C--   SET UP TO GET 4 VALUES
0112    1150  GASAVG=GASAVG+FLOAT(GASBUF(IAVG))
        D     TYPE 1151,IPEAK,GASAVG
0113    1151  FORMAT(/,1X,' IPEAK  GASAVG ',I5,5X,F10.2)
0114          RFVC=((GASAVG/4.)*SLOPE+YNTRCP)*BPTS
        CC-- !GET THE AVERAGE OF THE FVC
        C--  !AND NORMALIZE THE FVC.
        C--
        C--  GET THE FORCED EXPIRATORY VOLUMN IN 1 SECOND.
        C--
0115          FEV1=(FLOAT(GASBUF(100))*SLOPE+YNTRCP)*BPTS
        C--  !THE FEV1 IS EQUAL TO THE 100 TH.
        C--  !DATA WORD. NORMALEZED.
        C--
        C--  GET THE PEAK EXPIRATORY FLOW RATE (PEFR)
        C--
0116          PEFR=-2047
        C--  !SET THE INITIAL VALUE TO ZERO.
0117          DO 1200 IP=1,IPEAK
        C--  !ANALYZE 1 SEC. OF DATA.
0118          TAN=FLOAT(GASBUF(IP+4)-GASBUF(IP))/0.04
        C--  !CALCULATE THE TANGANT FOR EACH TIME INTERVAL.
```

C-2

```
        C--   !1 DATA POINT REPRESENTS 10 MSEC. REAL TIME.
0119           IF(TAN.GT.PEFR)PEFR=TAN
        C--   !FIND THE MAXIMUM TANGANT VALUE,
        C--   !THIS IS THE PEFR.
0121    1200   CONTINUE
0122           PEFR=(PEFR*SLOPE+YNTRCP)*BPTS
        C--   !STORE THE PEFR VALUE IN THE CORRECT BUFFER LOCATION.
        C--
        C--   GET THE FORCED EXPIRATORY FLOW BETWEEN 25% AND 75% MAX FVC.
        C--   (FORMERLY MMEF)
        C--
        C--   FEF 25% - 75%
        C--
0123           ITI75=1
0124           ITI25=1
        C--   !SET INITIAL TIMES FOR THE 25% AND 75%
        C--   !VALUES TO 0
0125           FVC75=(RFVC*0.75)
        C--   !CALCULATE 75% OF FVC.
0126           FVC25=(RFVC*0.25)
        C--   !CALCULATE 25% OF FVC.
0127           DO 1300 IF=1,IPEAK
        C--   !SET UP TO ANALYZE 1 SEC. CF TIME.
0128           RNUM=FLOAT(GASBUF(IF))*SLOPE+YNTRCP
0129           IF(RNUM.LE.FVC75)ITI75=IF
        C--   !GET THE TIME WHEN THE FVC IS 75% MAXIMUM.
0131           IF(RNUM.LE.FVC25)ITI25=IF
        C--   !GET THE TIME WHEN THE FVC IS 25% MAXIMUM.
0133    1300   CONTINUE
        D      TYPE 1301,FVC75,FVC25,ITI75,ITI25
0134    1301   FORMAT(/,1X,' FVC75  FVC25 ',2(2X,F10.2),
              %/,1X,' ITI75  ITI25 ',2(7X,I5))
0135           FEF=(FLOAT(GASBUF(ITI75)-GASBUF(ITI25))*SLOPE)/
              %(FLOAT(ITI75-ITI25)*0.01)
0136           FEF=FEF*BPTS
        C--   !CALCULATE THE FEF 25% - 75%
        C--   !BY CALCULATING THE SLOPE.
        C--   !STORE THE EFE VALUE IN THE CORRECT BUFFER LOCATION.
        C--
        C--   GET THE MEANS FORCED EXPIRATORY FLOW BETWEEN 200 ML. AND 12000 ML.
        C--   OF THE FORCED VITAL CAPACITY, (FEF 200 - 1200)
        C--   (FORMERLY MEFR).
0137           ITI12=1
0138           ITI2=1
        C--   !SET THE INITIAL TIMES FOR THE FVC'S AT 200
        C--   !AND 1200 MLS. TO 0.
0139           DO 1400 ITI=1,IPEAK
        C--   !SET UP TO ANALYZE 1. SEC OF REAL TIME.
0140           RNUM=FLOAT(GASBUF(ITI))*SLOPE+YNTRCP
0141           IF(RNUM.LE.0.2)ITI2=ITI
        C--   !GET THE TIME WHEN THE FVC IS EQUAL TO 200 ML.
0143           IF(RNUM.LE.1.2)ITI12=ITI
        C--   !GET THE TIME WHEN THE FVC IS EQUAL TO 1200 ML.
0145    1400   CONTINUE
```

```
        D     TYPE 1401,ITI12,ITI2
 0146   1401  FORMAT(/,1X,' ITI12  ITI2 ',2(5X,I5))
 0147         FEF212=(FLOAT(GASBUF(ITI12)-GASBUF(ITI2))*SLOPE)/
             %(FLOAT(ITI12-ITI2)*0.01)
 0148         FEF212=FEF212*8PTS
        C--  !CALCULATE THE FEF 200 - 1200 MLS.
        C--  !BY CALCULATING THE SLOPE BETWEEN THEM.
        C--  !STORE THR FEF212 VALUE IN THE CORRECT BUFFER LOCATION.
        C--
        C--   PREDICTED FORCED VITAL CAPACITY AND FORCED VITAL
        C--  CAPACITY AT 1 MIN. ARE CALCULATED USING THE FORMULAS REFERENCED
        C--    BY:
        C--      REUBEN M. CHERNIACK, MD.
        C--     IN:
        C--       PULMONARY FUNCTION TESTING
        C--       W.B. SAUNDERS, PP. 243, 1977.
        C--
        C--  GET THE % FEV1/FVC
        C--
 0149         F1FP=FEV1/RFVC*100.0
        C--  !CALCULATE THE % FEV1/FVC
        C--  !STORE THE F1FP IN THE CORRECT BUFFER LOCATION
        C--
        C--  GET THE PREDICTED FVC
        C--
 0150         IF(SEX.EQ.M)PFVC=0.06584*HIGHT-0.02954*AGE-5.12451
        C--  !FORMULA FOR CALCULATING THE
        C--  !PREDICTED FVC FOR MALES.
 0152         IF(SEX.EQ.F)PFVC=0.04071*HIGHT-0.02147*AGE-2.56958
        C--  !FORMULA FOR CALCULATING THE
        C--  !PREDICTED FVC FOR FEMALES.
        C--  !STORE THE PFVC VALUE IN THE CORRECT BUFFER LOCATION.
        C--
        C--  GET THE PERCENT OF THE OBSERVED FVC TO THE PREDICTED FVC
        C--
 0154         OPFVCP=RFVC/PFVC*100.0
        C--  !CALCULATE THE RATIO % OF THE
        C--  !OBSERVED TO PREDICTED FVC.
        C--  !STORE THE OPFVCP VALUE IN THE CORRECT BUFFER LOCATION.
        C--
        C--  GET THE PREDICTED FEV1.
        C--
 0155         IF(SEX.EQ.M)PFEV1=0.0425*HIGHT-0.03509*AGE-2.59946
        C--  !FORMULA FOR CACLLATING THE
        C--  !PREDICTED FEV1 FOR MALES.
 0157         IF(SEX.EQ.F)PFEV1=0.04071*HIGHT-0.02147*AGE-2.56958
        C--  !FORMULA FOR CALCULATING THE
        C--  !PREDICTED FEV1 FOR FEMALES.
        C--  !STORE THE PFEV1 VALUE IN THE CORRECT BUFFER LOCATION.
        C--
        C--  GET THE PERCENT OBSERVED FEV1 TO THE PREDICTED FEV1.
        C--
 0159         OPFEV1=FEV1/PFEV1*100.0
        C--  !CALCULATE THE % OF THE OBSERVED
```

```
      C--  !FEV1 DIVIDED BY THE PREDICTED FEV1.
      C--  !STORE THE OPFEV1 VALUE IN THE CORRECT BUFFER LOCATION.
      C--
      C--  PRINT OUT REPORT
      C--
0160        IMM=2*(DATE(1))
0161        IMM1=IMM-1
0162        TYPE 2000,CPID,(UNIQNO(I),I=1,2),RETNO,DATE(2),
           @MD(IMM1),MD(IMM),DATE(3)
0163        TYPE 2050,(NAME(I),I=1,14),SEX,(SSNO(I),I=1,3)
0164  2000  FORMAT(//32X,'FVC REPORT',/36X,'CCAS - V2.0',//
           %5X,'CPID NO.:  ',I5,28X,'UNIQUE NO.: ',A2,I4,//,
           %5X,'RETEST NO.:    ',I2,28X,'DATE:       ',I2,1X,2A2,I2,//)
0165  2050  FORMAT(5X,'SUBJECT NAME:  '14A2,1X,'SEX:           ',A2,//,
           %5X,'SOC. SEC. NO.: ',I3,'-',I2,'-',I4)
0166        TYPE 2100,RFVC,FEV1,PEFR,FEF,FEF212,F1FP,PFVC,OPFVCP,
           %PFEV1,OPFEV1
0167  2100  FORMAT(////,5X,'FVC (L)           ',F8.2,//,
           %5X,'FEV1 (L)          ',F8.2,//,
           %5X,'PEFR (L/SEC)      ',F8.2,//,
           %5X,'MMFR (L/SEC)      ',F8.2,//,
           %5X,'MEFR (L/SEC)      ',F8.2,//,
           %5X,'% FEV1/FVC        ',F8.2,//,
           %5X,'PRED FVC (L)      ',F8.2,//,
           %5X,'% OBS/PRED FVC    ',F8.2,//,
           %5X 'PRED FEV1 (L)     ',F8.2,//,
           %5X,'% OBS/PRED FEV1   ',F8.2,///,
           %5X,'ALL PFT DATA BTPS',/////////////////////////////)
0168  2200  CALL LIGHT("103)               !SET LIGHTS TO IDLE STATUS
0169        RETURN
0170        END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

M        000014  INTEGER*2 VARIABLE
F        000016  INTEGER*2 VARIABLE
PCSR     000024  INTEGER*2 VARIABLE
PCCB     000026  INTEGER*2 VARIABLE
PCS      000032  INTEGER*2 VARIABLE
PCC      000030  INTEGER*2 VARIABLE
OLD      001560  INTEGER*2 VARIABLE
DROUTB   000020  INTEGER*2 VARIABLE
DRINB    000022  INTEGER*2 VARIABLE
IFIX     000000  INTEGER*2 PROCEDURE
IPOKE    000000  INTEGER*2 PROCEDURE
INITI    000000  INTEGER*2 PROCEDURE
LIGHT    000000  INTEGER*2 PROCEDURE
TI       001562  REAL*4    VARIABLE
SECNDS   000000  REAL*4    PROCEDURE
DELTA    001566  REAL*4    VARIABLE
GASAVG   001572  REAL*4    VARIABLE
IPKVL    001576  INTEGER*2 VARIABLE
IPK      001600  INTEGER*2 VARIABLE
IPEAK    001602  INTEGER*2 VARIABLE
IAVG     001604  INTEGER*2 VARIABLE
FLOAT    000000  REAL*4    PROCEDURE
IP       001606  INTEGER*2 VARIABLE
TAN      001610  REAL*4    VARIABLE
ITI75    001614  INTEGER*2 VARIABLE
ITI25    001616  INTEGER*2 VARIABLE
FVC75    001620  REAL*4    VARIABLE
FVC25    001624  REAL*4    VARIABLE
IF       001630  INTEGER*2 VARIABLE
RNUM     001632  REAL*4    VARIABLE
ITI12    001636  INTEGER*2 VARIABLE
ITI2     001640  INTEGER*2 VARIABLE
ITI      001642  INTEGER*2 VARIABLE
IMM      001644  INTEGER*2 VARIABLE
IMM1     001646  INTEGER*2 VARIABLE
I        001650  INTEGER*2 VARIABLE


COMMON BLOCK /PDATA/    LENGTH 006000

IDATA    000000  INTEGER*2 ARRAY (1536)
SLOPE    000250  REAL*4    VARIABLE
YNTRCP   000254  REAL*4    VARIABLE
SSNO     000004  INTEGER*2 ARRAY (3)
UNIGNO   000014  INTEGER*2 ARRAY (2)
RETNO    000020  INTEGER*2 VARIABLE
CPID     000022  INTEGER*2 VARIABLE
DATE     000116  INTEGER*2 ARRAY (3)
NAME     000062  INTEGER*2 ARRAY (14)
SEX      000002  INTEGER*2 VARIABLE
AGE      000060  INTEGER*2 VARIABLE
HIGHT    000142  REAL*4    VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

BPTS     000224  REAL*4     VARIABLE
RFVC     000400  REAL*4     VARIABLE
FEV1     000404  REAL*4     VARIABLE
PEFR     000410  REAL*4     VARIABLE
FEF      000414  REAL*4     VARIABLE
FEF212   000420  REAL*4     VARIABLE
F1FP     000430  REAL*4     VARIABLE
PFVC     000440  REAL*4     VARIABLE
OPFVCP   000444  REAL*4     VARIABLE
PFEV1    000450  REAL*4     VARIABLE
OPFEV1   000454  REAL*4     VARIABLE

COMMON BLOCK /GASCOM/   LENGTH 007640

GASBUF   000000  INTEGER*2 ARRAY (2000)

COMMON BLOCK /ADCOM1/   LENGTH 000104

ADCTBF   000000  INTEGER*2 ARRAY (34)
GASDSP   000000  INTEGER*2 VARIABLE
EXHALE   000002  INTEGER*2 VARIABLE
SVTHRS   000004  INTEGER*2 VARIABLE
FVC      000006  INTEGER*2 VARIABLE
PUSHES   000014  INTEGER*2 VARIABLE
PFT      000016  INTEGER*2 VARIABLE
LFLSHR   000072  INTEGER*2 VARIABLE

COMMON BLOCK /X/        LENGTH 000060

MD       000000  INTEGER*2 ARRAY (24)
```

```
;*********************************************************************
;
;       SUBROUTINE ADISR
;
;       VERSION: 2              DATE: 08/24/79
;       AUTHOR: BILL CROSIER
;
;       PURPOSE:  SAMPLE SELECTED A/D CHANNELS,CHECK FRONT SWITCH PANEL
;                  (THRU PARALLEL INTERFACE), FLICKER OR FLASH SELECTED FRONT
;                PANEL LIGHTS, AND MAINTAIN ONE-SECOND AND 30-SECOND COUNTERS
;                FOR TIMING APPROPRIATE TEST ACTIVITIES.
;
;       SUMMARY OF ROUTINE:
;       ADISR IS CALLED EACH TIME THE KWV11-A PROGRAMMABLE CLOCK REQUESTS
;       AN INTERRUPT.  THE INTERRUPT RATE SHOULD BE SET BY THE MAIN
;       PROGRAM AT THE START OF EACH TEST AT 100 HZ.
;       EACH TIME ADISR IS CALLED, A SCAN
;       COUNTER (COUNT) IS DECREMENTED AND ANALOG CHANNEL 0 (SPIROMETER
;       VOLUME) IS SAMPLED AND COMPARED TO A THRESHOLD VOLUME (SVTHRS)
;       WHICH IS AN INTEGER VALUE PREVIOUSLY CALCULATED BY THE MAIN
;       PROGRAM WHICH CORRESPONDS TO A PARTICULAR AMOUNT OF SPIROMETER
;       DISPLACEMENT SUCH AS 0.2 LITERS.  IF THE SAMPLED VOLUME IS LESS
;       THAN THE THRESHOLD, NOTHING IS DONE WITH THE SAMPLE, AND THE ROUTINE
;       STORES 0 IN THE EXHALATION FLAG (EXHALE) AND THEN GOES TO READ
;       THE FRONT PANEL SWITCHES.
;       HOWEVER, IF THE SAMPLED VOLUME IS GREATER THEN OR EQUAL TO THE
;       THRESHOLD, "EXHALE" IS SET TO -1, THE SAMPLE IS STORED IN THE
;       GAS BUFFER (GASBUF) AND ITS POINTER IS INCREMENTED. (THE POINTER
;       IS GENERATED BY ADDING THE STARTING ADDRESS OF GASBUF TO
;       A RELATIVE DISPLACEMENT, GASDSP, ACCESSIBLE TO THE MAIN PROGRAM.
;       AT THE END OF THE GAS SAMPLING THE UPDATED VALUE OF GASDSP IS
;       STORED AGAIN. THIS ELIMINATES THE NEED TO ADD AN INDEX VALUE
;       TO THE BUFFER STARTING ADDRESS FOR EACH GAS VALUE.)
;                IF AN FVC IS NOT IN PROGRESS, THE PROGRAM THEN SAMPLES
;       ANALOG CHANNELS 1 (O2), 2 (N2), AND 3 (CO2), AND STORES EACH
;       SAMPLE IN SUCCESSIVE LOCATIONS IN GASBUF. AFTERWARDS, THE UPDATED
;       GAS BUFFER DISPLACEMENT POINTER IS STORED IN GASDSP. IT THEN
;       POINTS TO THE NEXT AVAILABLE LOCATION IN GASBUF.
;                NEXT, THE SCAN COUNTER IS CHECKED TO SEE IF IT
;       EGUALS 0. IF NOT, THE PROGRAM
;       RESTORES THE REGISTERS AND RETURNS. IF COUNT IS = 0, THEN A
;       NUMBER OF ACTIVITIFS TAKE PLACE AS FOLLOWS:
;                1. THE A/D IS TRIGGERED TO SAMPLE CH. 4 (HR).
;                2. COUNT IS RE-INITIALIZED TO 10.  THIS DETERMINES HOW
;                   FREQUENTLY (RELATIVE TO THE INTERRUPT RATE) ALL OF
;                   THESE OTHER ACTIVITIES TAKE PLACE.
;
;                3. THE FRONT PANEL SWITCHES ARE READ, AND BITS FOR
;                   SWITCHES THAT WERE JUST PUSHED, WHICH WERE NOT
;                   PREVIOUSLY RECOGNIZED BY THE MAIN PROGRAM, ARE SET
;                   IN "PUSHES".  THE MAIN PROGRAM SHOULD ACKNOWLEDGE
;                   RECEIPT OF THIS INFORMATION BY SAVING THE CONTENTS
;                   OF PUSHES AND THEN IMMEDIATELY CLEARING
;                   PUSHES BY STORING 0 IN IT.
;                4. THE PFT FLAG IS CHECKED AND IF NOT=0, THE PROGRAM
;                   BRANCHES TO END.
;                5. HEART RATE (HR) IS SAMPLED AND IF IT IS GREATER THAN
;                   HRTHRS, AN INTEGER VALUE CORRESPONDING TO A LOWER
;                   HEART RATE THRESHOLD AND PREVIOUSLY DETERMINED BY THE
;                   MAIN PROGRAM, THEN THE SAMPLED VALUE IS ADDED TO THE
;                   CURRENT HR (CHR) AND AVERAGE HR (AHR) ACCUMULATORS, AND THEIR
;                   CORRESPONDING COUNTERS (CHRCNT & AHRCNT) ARE INCREMENTED.
;                6. WORK LOAD (WL) OR ELEVATION (ELEV) AND SPEED ARE NEXT
;                    SAMPLED. THE SAMPLED DATA IS ADDED TO THE APPROPRIATE
;                    ACCUMULATORS (CWL & CSPD) AND THE WL-ELEV-SPEED COUNTER
;                    (CWLCNT) IS INCREMENTED.
;                 7. BPTIME, THE BP TIMER, IS DECREMENTED, AND AT THE START OF
;                    THE 55-TH SECOND OF EACH MINUTE AN INTERNAL BP COUNTER
;                    IS SET TO COLLECT SBP AND DBP SAMPLES DURING THE NEXT
;                    10 PASSES THRU THIS PORTION OF ADISR (EVERY 0.1 SEC).
;                    IN ORDER TO WORK PROPERLY, BPTIME MUST BE INITIALIZED BY
;                    THE MAIN PROGRAM AT THE BEGINNING OF THE EXERCISE TEST,
;                    JUST BEFORE THE PROGRAMMABLE CLOCK IS STARTED, TO
;                    10*(55-SECS), WHERE SECS IS THE NUMBER OF SECONDS PAST THE
;                    BEGINNING OF THE CURRENT MINUTE. OTHERWISE THE BLOOD
;                    PRESSURE CHANNELS WILL NOT BE SAMPLED AT THE CORRECT TIME.
;                        THE SBP AND DBP VALUES ARE ADDED TO THEIR RESPECTIVE
;                    ACCUMULATORS, CSBP AND CDBP. THE MAIN PROGRAM SHOULD
;                    DIVIDE THE SUMS BY 10 (RATHER THAN BY A VARIABLE COUNTER
;                    AS FOR HR,WL-ELEV, AND SPEED) IN ORDER TO GET THE AVERAGE
;                    BLOOD PRESSURES FOR THE MINUTE.
;                 8. AFTER SPEED IS SAMPLED (AND CONDITIONALLY, SBP & DBP),
;                    A 0.1 SECOND COUNTER IS DECREMENTED, AND EVERY 1 SECOND
;                    SECS IS INCREMENTED BY ONE. ANOTHER 1-SECOND COUNTER IS
;                    ALSO DECREMENTED, AND EVERY 30 SECS, SEC30 IS INCREMENTED
;                    BY 1. SECS & SECS30 MAY BE USED BY OTHER ROUTINES TO TIME
;                    OTHER ACTIVITIES AND MAY EITHER BE LEFT TO RUN CONTIN-
;                    UOUSLY, OR THEY MAY BE RESET TO ZERO PERIODICALLY BY OTHER
;                    ROUTINES. THESE TIMERS WILL NOT WORK PROPERLY UNLESS THE
;                    KWV11-A IS SET TO PROVIDE INTERRUPTS EVERY 10 MSEC.
;
;        NOTES:   1. THE ACCUMULATORS FOR CURRENT HR (CHR), AVERAGE HR (AHR),
;                    WL-ELEV (CWL), SPEED (CSPD), SBP (CSBP), AND DBP (CDBP)
;                    ARE DOUBLE-WORD INTEGERS, ALTHOUGH THEIR COUNTERS ARE
;                    SINGLE-WORD INTEGERS.  EACH ACCUMULATOR AND ITS RESPECT-
;                    IVE COUNTER (EXCEPT FOR SBP & DBP) MUST BE RESET TO 0 BY
;                    THE MAIN PROGRAM EACH TIME ONE DESIRES TO START A NEW
;                    AVERAGING INTERVAL. THIS IS NORMALLY DONE EACH TIME SECS30
;                    IS INCREMENTED, EXCEPT FOR CURRENT HR, WHICH MAY BE RESET
;                    EACH TIME SECS IS INCREMENTED.
;                 2. GENERAL REGISTERS USED:
;                        R0 -- POINTS TO A/D COMMAND & STATUS REGISTER
;                        R1 -- POINTS TO ONE OF SAMP0 THRU SAMP8 (A/D TRIGGER
;                              COMMAND WORDS)
;                        R2 -- POINTER FOR SAMPLED A/D DATA AND OTHER CALCULATE
;                              PARAMETERS
;                        R3 -- USED FOR SCAN COUNTER (RESET EACH TIME FRONT PANEL
;                              SWITCHES AND SLOW A/D SAMPLES ARE READ),
;                              ACKNOWLEDGED FRONT PANEL SWITCHS, AND SIGN
;                              EXTENSIONS OF A/D SAMPLES FOR HR,WL-ELEV,
;                              SPEED, SBP, AND DBP.
;                        R4 -- CONTAINS A/D SAMPLED DATA TEMPORARILY
;                              (EXCEPT O2,N2, AND CO2), AND FRONT PANEL
;                              SWITCH CONDITIONS
;                        R5 -- CONTAINS MASK FOR CHECKING A/D DONE FLAG.
;
;*****************************************************************************
```

```
                                .TITLE  ADISR
                        ;       PROGRAM:        ADISR.MAC
                        ;       VERSION:        2.0
                        ;       ANALOG TO DIGITAL CONVERTER INTERRUPT SERVICE ROUTINE FOR CDAS
                        ;       AUTHOR: WILLIAM G. CROSIER
                        ;       DATE:   24 AUG 79
                        ;
                        ;
                        ;
                                .MCALL  .REGDEF,..V2..
 00000                          ..V2..
 00000                          .REGDEF
        176770                  ACSR=176770                     ; A/D CONTROL & STATUS REGISTER
        176772                  ACIN=176772                     ; A/D INPUT DATA BUFFER
        176760                  DAC1=176760                     ; D/A #1 OUTPUT DATA BUFFER
        176762                  DAC2=176762                     ; D/A #2 OUTPUT DATA BUFFER
        167772                  DPOLTS=167772                   ; FRONT PANEL OUTPUT BUFFER (LIGHTS)
        167774                  DPINS=167774                    ; FRONT PANEL INPUT BUFFER (SWITCHES)
        172540                  CLKCSR=172540                   ; PROGRAMMABLE CLOCK CONTROL & STATUS REG.
 00000  010046  ADISR:          MOV     R0,-(SP)                ; SAVE REGISTERS ON STACK
 00002  010146                  MOV     R1,-(SP)                ;
 00004  012700                  MOV     #ACSR,R0                ; PUT ADDRESS OF A/D CSR IN R0
        176770
 00010  012701                  MOV     #SAMPC,R1               ; PUT A/D TRIGGER WORD POINTER IN R1
        000000'
 00014  012110                  MOV     (R1)+,(R0)              ; TRIGGER A/D FOR CHAN. 0 (SPIR. VOL.)
 00016  010246                  MOV     R2,-(SP)                ; SAVE OTHER REGISTERS
 00020  010346                  MOV     R3,-(SP)
 00022  010446                  MOV     R4,-(SP)
 00024  010546                  MOV     R5,-(SP)
 00026  012705                  MOV     #200,R5                 ; STORE A/D DONE MASK IN R5
        000200
 00032  012702                  MOV     #GASBUF,R2              ; PUT GAS DATA BUFFER STARTING ADR. IN R2
        000000'
 00036  016703                  MOV     GASDSP,R3               ; PUT GAS DISPLACEMENT POINTER (WORD COUNT) IN R3
        000000'
 00042  020327                  CMP     R3,#2000.               ;SEE IF GAS DATA BUFFER IS FULL
        003720
 00046  100405                  BMI     GETDSP                  ;IF NOT, GO CALCULATE DISPLACEMENT POINTER.
 00050  005003                  CLR     R3                      ;RESET R3 TO 0.
 00052  005067                  CLR     GASDSP                  ;RESET GAS DISPLACEMENT TO 0.
        000000'
 00056  005267                  INC     GASOVF                  ;INCREMENT GAS DISPLACEMENT OVERFLOW FLAG.
        000074'
 00062  006303  GETDSP:         ASL     R3                      ; MULT. R3 BY 2 (CONVERT WORD TO BYTE COUNT)
 00064  060302                  ADD     R3,R2                   ; & ADD REL. DISPLACEMENT TO GENERATE POINTER
 00066  016703                  MOV     COUNT,R3                ; LOAD SCAN COUNTER
        000000'
 00072  005303                  DEC     R3                      ; DECREMENT COUNTER
 00074  010367                  MOV     R3,COUNT                ; & STORE IT (LEAVE IT IN R3 FOR LATER USE)
        000000'
 00100  031005  LOOP0:          BIT     (R0),R5                 ; SEE IF A/D CONVERSION IS DONE
 00102  001776                  BEQ     LOOP0                   ; IF NOT, WAIT
 00104  013704                  MOV     @#ACIN,R4               ; PUT A/D SAMPLE IN R4
        176772
 00110  005767                  TST     EXHALE                  ; EXHALATION IN PROGRESS DURING LAST SAMPLE?
        000002'
```

```
46 00114 001431            BEQ     NOEXH             ; IF NOT, GO TO NOEXH
47 00116 020467            CMP     R4,SVTHRS         ; IS SAMPLE .GT. THRESHOLD?
         000004'
48 00122 100013            BPL     HIGH1             ; IF SO, GO RESET LOCNT & CHECK FVC FLAG
49 00124 005367            DEC     LOCNT             ; IF SAMPLE .LE. THRESHOLD, DECREMENT LOCNT
         000570
50 00130 002013            BGE     FVCTST            ; IF LOCNT IS STILL .GE. 0, GO TO FVCTST
51 00132 005067            CLR     EXHALE            ; OTHERWISE, SET EXHALE FLAG = 0 (FALSE)
         000002'
52 00136 005267            INC     EXHCNT            ; INCREMENT END OF BREATH EXHALATION COUNTER
         000012'
53 00142 012767            MOV     #5,LOCNT          ; RESET LOCNT TO 5
         000005
         000550
54 00150 000456            BR      READSW            ; GO READ FRONT PANEL SWITCHES
55 00152 012767 HIGH1:     MOV     #5,LOCNT          ; RESET LOCNT TO 5
         000005
         000540
56 00160 005767 FVCTST:    TST     FVC               ; IS THIS AN FVC TEST?
         000006'
57 00164 001430            BEQ     EXH               ; IF NOT, GO TO EXH
58 00166 010422 STORSV:    MOV     R4,(R2)+          ; IF FVC, STORE SP. VOL. SAMPLE IN GAS BUFFER
59 00170 012767            MOV     #-1,EXHALE        ; SET EXHALE = -1 (TRUE)
         177777
         000002'
60 00176 000443            BR      READSW            ; GO READ FRONT PANEL SWITCHES
61 00200 020467 NOEXH:     CMP     R4,SVTHRS         ; IS SAMPLE .GT. THRESHOLD?
         000004'
62 00204 100004            BPL     HIGH              ; IF SO, GO TO HIGH
63 00206 012767            MOV     #5,HICNT          ; OTHERWISE, RESET HICNT TO 5
         000005
         000506
64 00214 000434            BR      READSW            ; GO READ FRONT PANEL SWITCHES
65 00216 005767 HIGH:      TST     FVC               ; IS THIS AN FVC TEST?
         000006'
66 00222 001361            BNE     STORSV            ; IF SO, GO STORE SPIR. VOL. SAMPLE
67 00224 005367            DEC     HICNT             ; DECREMENT HICNT
         000472
68 00230 002026            BGE     READSW            ; IF HICNT IS STILL .GE. 0, GO TO READSW
69 00232 012767            MOV     #-1,EXHALE        ; OTHERWISE, SET EXHALE = -1 (TRUE)
         177777
         000002'
70 00240 012767            MOV     #5,HICNT          ; RESET HICNT TO 5
         000005
         000454
71 00246 010422 EXH:       MOV     R4,(R2)+          ; STORE SPIRO. VOLUME SAMPLE IN GAS BUFFER
72 00250 012110            MOV     (R1)+,(R0)        ; TRIGGER A/D FOR CH. 1 (O2)
73 00252 031005 LOOP1:     BIT     (R0),R5           ; SEE IF CONVERSION DOONE
74 00254 001776            BEQ     LOOP1             ; IF NOT WAIT
75 00256 013722            MOV     @#ADIN,(R2)+      ; STORE A/D DATA (O2)
         176772
76 00262 012110            MOV     (R1)+,(R0)        ; TRIGGER A/D FOR CH. 2 (N2)
77 00264 031005 LOOP2:     BIT     (R0),R5           ; SEE IF CONVERSION DONE
78 00266 001776            BEQ     LOOP2             ; IF NOT WAIT
79 00270 013722            MOV     @#ADIN,(R2)+      ; STORE A/D DATA (N2)
         176772
80 00274 012110            MOV     (R1)+,(R0)        ; TRIGGER A/D FOR CH. 3 (CO2)
```

```
ADIBR   RT-11 MACRO VM02-12   29-FEB-80 14:22:36 PAGE 1+

81 00276 031005 LOOP3:  BIT     (R0),R5         ; SEE IF CONVERSION DONE
82 00300 001776         BEQ     LOOP3           ; IF NOT WAIT
83 00302 013722         MOV     @#ACIN,(R2)+    ; STORE A/D DATA (CO2)
         176772
84 00306 012701 READSW: MOV     #SAMP4,R1       ; PUT CH. 4 A/D TRIGGER WORD POINTER IN R1
         000010'
85 00312 162702         SUB     #GASBUF,R2      ; SUBTRACT GAS BUF. STARTING ADDR. FROM
         000000'
86                                              ; GAS POINTER
87 00316 006202         ASR     R2
88 00320 010267         MOV     R2,GASDSP       ; STORE NEW GAS DATA BUFFER DISPLACEMENT
         000000'
89 00324 005703         TST     R3              ; SCAN COUNTER = 0 ?
90 00326 003165         BGT     FINISH          ; IF NOT, GO TO END
91 00330 012767         MOV     #10.,COUNT      ; RE-INITIALIZE SCAN COUNTER
         000012
         000000'
92                ;     MOV     @#DRINB,R4      ; READ FRONT PANEL SWITCHES
93 00336 012702         MOV     #PFT,R2         ; PUT DATA POINTER IN R2
         000016'
94                ;     MOV     #MASK,R2        ; PUT POINTER TO MASK IN R2
95                ;     BIC     (R2)+,R4        ; MASK OUT UNUSED BITS
96                ;     COM     R4              ; COMPLEMENT OF CURRENT SWITCH CONDITIONS
97                ;     MOV     (R2),R3         ; PUT PREVIOUS SWITCH CONDITIONS IN R3
98                ;     BIC     R4,R3           ; CLEAR BITS IN R3 FOR SWITCHES THAT
99                ;                             ; ARE OFF (NOT PUSHED).
100               ;     COM     R4              ; RESTORE CURRENT SWITCH CONDITIONS IN R4
101               ;     BIC     R3,R4           ; CLEAR BITS IN R4 FOR PREVIOUSLY RECOGNIZED
102               ;                             ; SWITCH PUSHES. NOW R4 HAS 1'S ONLY FOR NEWLY
103               ;                             ; PUSHED SWITCHES, SO:
104               ;                             ; R4=PRESENT SWITCHES.AND.(.NOT.PREVSW)
105               ;     BIS     R4,R3           ; SET BITS IN R3 FOR NEW SWITCH PUSHES,SO
106               ;                             ; R3 NOW HAS PRESENT SWITCH CONDITIONS
107               ;                             ; (SAME AS @#DRINB)
108               ;     MOV     R3,(R2)+        ; STORE PRESENT SWITCH CONDITIONS IN PREVSW
109               ;     BIS     R4,(R2)+        ; STORE NEW SWITCH PUSH INFO IN PUSHES
110 0342 005727         TST     (R2)+           ; IS PFT IN PROGRESS ?
111 0344 001136         BNE     FLICKR          ; IF SO, GO TO END
112 0346 012110         MOV     (R1)+,(R0)      ; TRIGGER A/D FOR CH. 4 (HR)
113 0350 005222         INC     (R2)+           ; INCREMENT CURRENT HR POINTER
114 0352 005222         INC     (R2)+           ; INCREMENT 30- SEC AVG. HR COUNTER
115 0354 031005 LOOP4:  BIT     (R0),R5         ; SEE IF A/D CONVERSION DONE
116 0356 001776         BEQ     LOOP4           ; IF NOT WAIT
117 0360 013704         MOV     @#ACIN,R4       ; PUT HR A/D SAMPLE IN R4
         176772
118 0364 006703         SXT     R3              ; EXTEND SIGN INTO R3
119 0366 012110         MOV     (R1)+,(R0)      ; TRIGGER A/D FOR CH. 5 (WL-ELEV)
120 0370 022204         CMP     (R2)+,R4        ; HR THRESHOLD .GT. SAMPLE ?
121 0372 100407         BMI     ADDHR           ; SKIP IF NOT
122 0374 005367         DEC     CHRCNT          ; DECREMENT HR COUNTERS SINCE HR
         000020'
123 0400 005367         DEC     AHRCNT          ; SAMPLE WAS INVALID (.LT. THRESHOLD).
         000022'
124 0404 012702         MOV     #CWL,R2         ; PUT POINTER TO CWL IN R2
         000036'
125 0410 000406         BR      LOOP5           ; GO READ WL-ELEV SAMPLED VALUE
126 0412 060422 ADDHR:  ADD     R4,(R2)+        ; ACCUMULATE CURRENT HR SUM
```

```
ADISP   RT-11 MACRO VM02-12   29-FEB-80 14:22:30 PAGE 14

127 0414 005512                ADC     (R2)              ; ADD CARRY
128 0416 060322                ADD     R3,(R2)+          ; ADD HIGH-ORDER PART OF CHR
129 0420 060422                ADD     R4,(R2)+          ; ACCUMULATE 30-SEC AVG. HR SUM
130 0422 005503                ADC     R3                ; ADD CARRY
131 0424 060322                ADD     R3,(R2)+          ; ADD HIGH-ORDER PART OF AHR
132 0426 031005 LOOP5:         BIT     (R0),R5           ; SEE IF A/D CONVERSION DONE
133 0430 001776                BEQ     LOOP5             ; IF NOT WAIT
134 0432 013704                MOV     @#ADIN,R4         ; PUT WL-ELEV A/D SAMPLE IN R4
         176772
135 0436 006703                SXT     R3                ; EXTEND SIGN INTO R3
136 0440 012110                MOV     (R1)+,(R0)        ; TRIGGER A/D FOR CH. 6 (SPEED)
137 0442 060467                ADD     R4,AWL            ;ACCUMULATE CURRENT WL SUM
         000076'
138 0446 005567                ADC     AWL1              ;ADD CARRY
         000100'
139 0452 060367                ADD     R3,AWL1           ;ADD HIGH-ORDER PART OF AWL
         000100'
140 0456 005267                INC     AWLCNT            ;INCREMENT CURRENT COUNTER
         000102'
141 0462 060422                ADD     R4,(R2)+          ; ACCUMULATE 30 SEC AVG WL-ELEV SUM
142 0464 005503                ADC     R3                ; ADD CARRY
143 0466 060322                ADD     R3,(R2)+          ; ADD HIGH-ORDER PART OF CWL
144 0470 005222                INC     (R2)+             ; INCREMENT 30 SEC AVG WL-ELEV COUNTER
145 0472 005322                DEC     (R2)+             ; DECREMENT BP TIMER
146 0474 100006                BPL     LOOP6             ; SKIP IF NOT TIME TO START BP READING
147 0476 012767                MOV     #600.,BPTIME      ; RESET BP TIMER TO 60 SEC = 10 SAMP/SEC
         001130
         000044'
148 0504 016767                MOV     BPCNT,BPCNT1      ; RESET BP COUNTER FOR BPCNT SAMPLES
         000064'
         000052'
149 0512 031005 LOOP6:         BIT     (R0),R5           ; SEE IF A/D CONVERSION DONE
150 0514 001776                BEQ     LOOP6             ; IF NOT WAIT
151 0516 013704                MOV     @#ADIN,R4         ; PUT A/D SAMPLE (SPEED) IN R4
         176772
152 0522 006703                SXT     R3                ; EXTEND SIGN IN TO R3
153 0524 060422                ADD     R4,(R2)+          ; ACCCUMULATE CURRENT SPEED SUM
154 0526 005503                ADC     R3                ; ADD CARRY
155 0530 060322                ADD     R3,(R2)+          ; ADD HIGH-ORDER PARTS
156 0532 005322                DEC     (R2)+             ; DECREMENT BP COUNTER
157 0534 100422                BMI     TIMER             ; SKIP IF ALREADY HAVE BPCNT SAMPLES
158 0536 012110                MOV     (R1)+,(R0)        ; TRIGGER A/D FOR CH. 7 (SBP)
159 0540 031005 LOOP7:         BIT     (R0),R5           ; SEE IF A/D CONVERSION DONE
160 0542 001776                BEQ     LOOP7             ; IF NOT WAIT
161 0544 013704                MOV     @#ADIN,R4         ; PUT SBP A/D SAMPLE IN R4
         176772
162 0550 006703                SXT     R3                ; EXTEND SIGN INTO R3
163 0552 012110                MOV     (R1)+,(R0)        ; TRIGGER A/D FOR CH. 8 (DBP)
164 0554 060422                ADD     R4,(R2)+          ; ACCUMULATE CURRENT SBP SUM
165 0556 005503                ADC     R3                ; ADD CARRY
166 0560 060322                ADD     R3,(R2)+          ; ADD HIGH-ORDER PARTS
167 0562 031005 LOOP8:         BIT     (R0),R5           ; SEE IF A/D CONVERSION DONE
168 0564 001776                BEQ     LOOP8             ; IF NOT WAIT
169 0566 013704                MOV     @#ADIN,R4         ; PUT SAMPLE (DBP) IN R4
         176772
170 0572 006703                SXT     R3                ; EXTEND SIGN INTO R3
171 0574 060422                ADD     R4,(R2)+          ; ACCUMULATE CURRENT DBP SUM
```

```
172 0576 005503            ADC     R3              ; ADD CARRY
173 0600 060322            ADD     R3,(R2)+        ; ADD HIGH-ORDER PARTS
174 0602 005367 TIMER:     DEC     CNT10           ; DECREMENT 0.1 SECOND COUNTER
         000002'
175 0606 003015            BGT     FLICKR          ; SKIP IF 1 SEC. HAS NOT ELAPSED
176 0610 012767            MOV     #10.,CNT10      ; RESET CNT10 FOR 10 COUNTS
         000012
         000002'
177 0616 005267            INC     SECS            ; INCREMENT SECONDS COUNTER/FLAG
         000066'
178 0622 005367            DEC     CNT30           ; DECREMENT 30-SEC COUNTER
         000004'
179 0626 001005            BNE     FLICKR          ; SKIP IF 30 SECS HAVE NOT ELAPSED
180 0630 012767            MOV     #30.,CNT30      ; RESET CNT30 FOR 30 COUNTS
         000036
         000004'
181 0636 005267            INC     SEC030          ; INCREMENT 30-SEC. COUNTER/FLAG
         000070'
182                                                ;
183 0642 005367 FLICKR:    DEC     FLRCNT          ; DECREMENT FLICKER COUNTER
         000006'
184 0646 003015            BGT     FINISH          ; SKIP IF NOT TIME TO FLICKER LIGHTS
185 0650 012767            MOV     #2,FLRCNT       ; RESET FLRCNT TO CHANGE LIGHT STATUS
         000002
         000006'
186                                                ; EVERY 0.2 SEC. (FLICKER RATE= 2.5HZ)
187 0656 005167            COM     FLRFLG          ; COMPLEMENT FLICKER FLAGS
         000010'
188 0662 100004            BPL     LITEON          ; EVERY OTHER TIME, GO TURN LIGHTS ON
189 0664 046737            BIC     LFLSHR,@#DROUTS ; TURN OFF APPROPRIATE LIGHTS
         000072'
         167772
190 0672 000403            BR      FINISH          ; GO TO END
191 0674 056737 LITEON:    BIS     LFLSHR,@#DROUTS ; TURN ON APPROPRIATE LIGHTS
         000072'
         167772
192                                                ;
193 0702 012605 FINISH:    MOV     (SP)+,R5        ;RESTORE REGISTERS
194 0704 012604            MOV     (SP)+,R4
195 0706 012603            MOV     (SP)+,R3
196 0710 012602            MOV     (SP)+,R2
197 0712 012601            MOV     (SP)+,R1
198 0714 012600            MOV     (SP)+,R0
199 0716 000002            RTI                     ; RETURN
200                                                ;
201 0720 000005 LOCNT:     .WORD   5
202 0722 000005 HICNT:     .WORD   5
203                                                ;
204      000000'           .CSECT TIMCNT
205 0000 000012 COUNT:     .WORD   10.             ; SCAN COUNTER
206 0002 000012 CNT10:     .WORD   10.             ; 0.1 SEC. COUNTER
207 0004 000036 CNT30:     .WORD   30.             ; 1 SEC. COUNTER
208 0006 000000 FLRCNT:    .WORD   0               ; FLICKER COUNTER (FOR LIGHTS)
209 0010 000000 FLRFLG:    .WORD   0               ; FLICKER FLAG
210 0012 000000 EXHCNT:    .WORD   0               ;EXHALATION COUNTER.
211                                                ;
212      000000'           .CSECT  ADCOM
```

```
213 0000 000020 SAMP0:  .WORD   20              ; CH. 0 (SP. VOL) A/D START COMMAND
214 0002 000420 SAMP1:  .WORD   420             ; CH. 1 (O2)
215 0004 001020 SAMP2:  .WORD   1020            ; CH. 2 (N2)
216 0006 001420 SAMP3:  .WORD   1420            ; CH. 3 (CO2)
217 0010 002020 SAMP4:  .WORD   2020            ; CH. 4 (HR)
218 0012 002420 SAMP5:  .WORD   2420            ; CH. 5 (WL-ELEV)
219 0014 003020 SAMP6:  .WORD   3020            ; CH. 6 (SPEED)
220 0016 003420 SAMP7:  .WORD   3420            ; CH. 7 (SBP)
221 0020 004020 SAMP8:  .WORD   4020            ; CH. 8 (DBP)
222                                             ;
223      000000'        .CSECT  ADCOM1
224 0000 000000 GASDSP: .WORD   0               ; GAS BUFFER DISPLACEMENT (SUBSCRIPT)
225 0002 000000 EXHALE: .WORD   0               ; FLAG (=0 FOR INHALATION,=-1 FOR
226                                             ; EXHALATION)
227 0004 000000 SVTHRS: .WORD   0               ; SPIROMETER VOL. THRESHOLD
228 0006 000000 FVC:    .WORD   0               ; FLAG (=-1 FOR FVC, 0 ALL OTHERS)
229 0010 177400 MASK:   .WORD   177400          ; MASKS OUT UNUSED BITS ON PARALLEL
230                                             ; INTERFACE INPUT LINES (UNUSED SWITCHES)
231 0012 000000 PREVSW: .WORD   0               ; CCONTAINS 1'S FOR SWITCHES THAT HAVE
232                                             ; ALREADY BEEN RECOGNIZED AS BEING PUSHED
233 0014 000000 PUSHES: .WORD   0               ; CONTAINS 1'S FOR BUTTONS THAT ARE PUSHED
234 0016 000000 PFT:    .WORD   0               ; FLAG (=-1 FOR ALL PFT'S, 0 OTHERWISE)
235 0020 000000 CHRCNT: .WORD   0
236 0022 000000 AMRCNT: .WORD   0
237 0024 000000 HRTHRS: .WORD   0
238 0026 000000 CHR:    .WORD   0
239 0030 000000 CHR1:   .WORD   0
240 0032 000000 AHR:    .WORD   0
241 0034 000000 AHR1:   .WORD   0
242 0036 000000 CAL:    .WORD   0
243 0040 000000 CAL1:   .WORD   0
244 0042 000000 CALCNT: .WORD   0
245 0044 001046 BPTIME: .WORD   550.
246 0046 000000 CSPC:   .WORD   0
247 0050 000000 CSPC1:  .WORD   0
248 0052 000000 BPCNT1: .WORD   0
249 0054 000000 CSBP:   .WORD   0
250 0056 000000 CSBP1:  .WORD   0
251 0060 000000 CDBP:   .WORD   0
252 0062 000000 CDBP1:  .WORD   0
253 0064 000012 HRCNT:  .WORD   10.
254 0066 000000 SECS:   .WORD   0
255 0070 000000 SECS30: .WORD   0
256 0072 000000 LFLSHR: .WORD   0
257 0074 000000 GASCVF: .WORD   0
258 0076 000000 AWL:    .WORD   0
259 0100 000000 AWL1:   .WORD   0
260 0102 000000 AWLCNT: .WORD   0
261                                             ;
262      000000'        .CSECT  GASCOM
263 0000        GASBUF: .BLKW   2000.           ; GAS DATA BUFFER
264                                             ;
265      000000         .ASECT
266      000104         .=104
267 0104 000000'        .WORD   ADISR           ;KW11-P INTERRUPTS VECTOR TO ADISR
268 0106 000200         .WORD   200             ;DISABLE ALL OTHER INTERRUPTS.
269      000001'        .END
```

```
ADISR   RT-11 MACRO VM02-12   29-FEB-80 14:22:30 PAGE 1+
SYMBOL TABLE

ADDMR   000412R         ACIN  = 176772          ADISR   000000R
ADSR  = 176770          AMR     000032R    004  AMRCNT  000022R    004
AMP1    000034R    004  AWL     000076R    004  AWLCNT  000102R    004
AWL1    000100R    004  BPCNT   000064R    004  BPCNT1  000052R    004
BPTIME  000044R    004  CCRP    00006CR    004  CCRP1   000062R    004
CHR     000026R    004  CHRCNT  00002CR    004  CHR1    000030R    004
CLKCSR= 172540          CNT10   000002R    002  CNT30   000004R    002
COUNT   000000R    002  CSRP    000054R    004  CSRP1   000056R    004
CSPC    000046R    004  CSPC1   000050R    004  CWL     000036R    004
CWLCNT  000042R    004  CWL1    000040R    004  DAC1  = 176760
DAC2  = 176762          DRINB = 167774          DROUTB= 167772
EXH     000246R         EXHALE  000002R    004  EXHCNT  000012R    002
FINISH  000702R         FLICKR  000642R         FLRCNT  000006R    002
FLRFLG  000010R    002  FVC     00000C6    004  FVCTST  000160R
GASBUF  000000R    005  GASCSR  00000CR    004  GASOVF  000074R    004
GETCSR  000062R         HICNT   000722R         HIGH    000216R
HIGH1   000152R         HRTHRS  000024R    004  LFLSHR  000072R    004
LITEON  000674R         LOCNT   000720R         LOOP0   000100R
LOOP1   000252R         LOOP2   000264R         LOOP3   000276R
LOOP4   000354R         LOOP5   000426R         LOOP6   000512R
LOOP7   000540R         LOOP8   000562R         MASK    000010R    004
NOEXH   000200R         PC    =%000007          PFT     000016R    004
PREVSW  000012R    004  PLSRES  000014R    004  READSW  000306R
R0    =%000000          R1    =%000001          R2    =%000002
R3    =%000003          R4    =%000004          R5    =%000005
SAMP0   000000R    003  SAMP1   000002R    003  SAMP2   000004R    003
SAMP3   000006R    003  SAMP4   000010R    003  SAMP5   000012R    003
SAMP6   000014R    003  SAMP7   000016R    003  SAMP8   000020R    003
SECS    000066R    004  SECS30  000070R    004  SP    =%000006
STORSV  000166R         SVTHRS  000004R    004  TIMER   000602R
...V2 = 000001
. ABS.  000110     000
        000724     001
TIMCNT  000014     002
ADCOM   000022     003
ADCOM1  000104     004
GASCOM  007640     005
ERRORS DETECTED: 0
FREE CORE: 17567. WORDS

.LP:=ADISR
```

```
        C--   EXERCIZE SUBROUTINE, VERSION 2.0
        C--
        C      WRITTEN BY: ROY A. REED
        C      DATE:       OCT. 1978
        C
        C      REWRITTEN BY: WILLIAM G. CROSIER
        C      DATE:          AUG. 24,1979
        C      MODIFIED BY: ABHIJIT GADGIL
        C      DATE:      DEC,1979
        C
0001           SUBROUTINE EX
        C--
        C--   SET UP VARIABLES
        C--
0002           LOGICAL*1 RETI(8), RETIX(8)
        C--   !CONTAINES THE REAL TIME
0003           INTEGER EXHCNT,TIMBUF(6)
0004           INTEGER SECS,BPCNT,BPCNT1,BPTIME,SECS30
        C--   !30 SECOND TIMER, FROM ADISR
0005           INTEGER*4 OUT
0006           INTEGER AGE
        C--   !PATIENTS AGE, TO BE USED IN
        C--   !THE CALCULATIONS OF THE PREDICTD VALUES.
0007           INTEGER CPID
        C--   !PATIENTS CARDIOPULMONARY IDENTITY NUMBER.
0008           INTEGER NAME(14)
        C--   !PATIENT'S NAME.
0009           INTEGER SSNO(3)
        C--   !PATIENT'S SOCIAL SECURITY NUMBER.
0010           INTEGER DATE(3)
        C--   !DATE THE TEST WAS  CONDUCTED.
0011           INTEGER RETNO
        C--   !PATIENT'S RETEST NUMBER.
0012           INTEGER UNIQNO(2)
        C--   !PATIENT'S UNIQUE NUMBER.
0013           INTEGER M,F,SEX
        C--   !SUBJECTS SEX TO BE USED IN CALCULATIONS.
0014           INTEGER ADCTBF(34)
        C--   !COMMON AREA FOR A/D COMMAND VARIABLES
0015           INTEGER GASBUF(2000)
        C--   !A/D COMMON DATA BUFFER.
0016           INTEGER GASDSP
        C--   !GAS BUFFER DISPLACEMENT
0017           INTEGER EXHALE
0018           INTEGER SVTHRS
        C--   !SPIROMETER VOLUME THRESHOLD.
0019           INTEGER FVC
        C--   !FLAG (=-1FOR FORCED VITAL CAPACITY
        C     !EQUAL TO 0 FOR ALL OTHERS).
0020           INTEGER PFT
        C--   !FLAG =-1 FOR PULMONARY FUNCTION TESTS
        C     !      =-0 FOR ALL OTHERS
0021           INTEGER HRTHRS
        C--   !CONTAINES THE HEART RATE THRESHOLD.
```

```
      C     !PUSHED.
0022         INTEGER PCSR
      C--   !ADDRESS OF THE PROGRAMABLE CLOCK
      C     !STATUS REGISTER.
0023         INTEGER PCCB
      C--   !ADDRESS OF THE PROGRAMABLE CLOCK
      C     !COUNTER BUFFER.
0024         INTEGER PCS
      C--   !HAS THE VALUE TO BE PLACED INTO THE
      C     !CLOCK STATUS REGISTER.
0025         INTEGER PCC
      C--   !HAS THE VALUE TO BE PLACED INTO THE
      C     !CLOCK COUNTER BUFFER.
0026         INTEGER OLD
      C--   !VARIABLE TO HOLD THE OLD VALUE OF EXHAST.
0027         INTEGER DROUTB,DRINB
      C--   !VARIBLES THAT CONTAIN THE ADDRESS OF THE
      C--   !DR11 PARALLEL INTERFACE CARD INPUT AND OUTPUT BUFFERS.
0028         REAL HIGHT
      C--   !PATIENT'S HEIGHT IN CM.
0029         REAL WT
0030         REAL SLPYCP(18)
      C--   !VARIEBLES THAT CONTAIN INFORMATION PASSED
      C--   !THROUGH ADCOM1 COMMON BLOCK TO CALCULATE THE
      C--   !SPIROMETER THRESHOLD (SLOPE & Y-INTERCEPT
      C--   !ARRAY -- CALIB. FACTORS FOR ALL 9 CHANNELS
0031         REAL*8 SECTN,RST,EX,REC      !ALPHA LABELS FOR SECTION OF TEST
      C--   SET UP THE COMMON AREAS.
      C--
0032         COMMON/GAS30/VC2,VCC2,VOL,IBRTH
0033         COMMON/TIMCNT/TIMBUF
0034         COMMON/PDATA/IDATA(1536)
      C--   !SET UP THE COMMON BUFFER FOR PATIENT DATA
      C--   !ALSO, CONTAINES THE PATIENTS INFORMATION.
0035         COMMON /GASCOM/GASBUF
      C--   !DATA BUFFER COMMON AREA.
0036         COMMON /ADCOM1/ADCTBF
      C--   !COMMON AREA TO PASS COMMAND VARIABLES AND
      C     !CONSTANTS TO &FROM A/D INTERRUPT SERVICE ROUTINE
0037         EQUIVALENCE (TIMBUF(6),EXHCNT)
      C--   !CONTAINES THE END OF BREATH FLAG.
0038         EQUIVALENCE (IDATA(85),SLPYCP(1))
      C--   ! CONTAINES THE SLOPE AND Y INTERCEPTS.
0039         EQUIVALENCE(IDATA(3),SSNO(1))
      C--   !THESE THREE DATA WORDS CONTAIN
      C--   !THE PATIENT'S SOCIAL SECURITY NUMBER.
0040         EQUIVALENCE (IDATA(7),UNIQNO)
      C--   !THIS DATA WORD  CONTAINES THE PATIENT'S
      C--   !UNIQUE NUMBER.
0041         EQUIVALENCE (IDATA(9),RETNO)
      C--   !THIS DATA WORD CONTAINES THE SUBJECT'S
      C--   !RETEST NUMBER.
0042         EQUIVALENCE (IDATA(10),CPIC)
      C--   !THIS DATA WORD CONTAINES THE PATIENT'S
      C--   !CARDIOPULMUNARY NUMBER.
0043         EQUIVALENCE(IDATA(40),DATE(1))
      C--   !THESE THREE DATA WORDS CONTAIN THE
      C--   !PATIENT'S TEST DATE.
0044         EQUIVALENCE (IDATA(26),NAME(1))
      C--   !THE 14 DATA WORDS CONTAIN THE
      C--   !PATIENT'S NAME.
0045         EQUIVALENCE (IDATA(2),SEX)
      CC    !THIS DATA WORD CONTAINES THE PATIENTS SEX.
0046         EQUIVALENCE (IDATA(25),AGE)
      C--   !THIS DATA WORD CONTAINES THE PATIENTS AGE.
0047         EQUIVALENCE (IDATA(50),HIGHT)
      C--   !THIS DATA WORD CONTAINES THE PATIENTS HEIGHT.
0048         EQUIVALENCE (IDATA(52),WT)
0049         EQUIVALENCE (IDATA(72),KCNT)
0050         EQUIVALENCE (ADCTBF(19),BPTIME),(ADCTBF(22),BPCNT1)
0051         EQUIVALENCE (ADCTBF(27),BPCNT)
0052         EQUIVALENCE(ADCTBF(1),GASDSP)
      C--    GAS BUFFER DISPLACEMENT
0053         EQUIVALENCE (ADCTBF(2),EXHALE)
0054         EQUIVALENCE (ADCTBF(3),SVTHRS)
      C--   !SPIROMETER VOLUMN THRESHOLD, MUST BE CALCULATED.
0055         EQUIVALENCE(ADCTBF(4),FVC)
      C--   !FLAG EQUAL TO -1 FOR FORCED VITIAL CAPACTTY.
0056         EQUIVALENCE (ADCTBF(7),PUSHES)
      C--   !CONTAINES 1'S FOR BUTTONS THAT ARE
      C     !PUSHED BY OPERATOR.
0057         EQUIVALENCE (ADCTBF(8),PFT)
      C--   !FLAG FOR THE PFT TEST, MUST BE SET TO -1 FOR PFT'S
0058         EQUIVALENCE (ADCTBF(11),HRTHRS)
0059         EQUIVALENCE (ADCTBF(16),ML),(ADCTBF(20),AS)
0060         EQUIVALENCE (ADCTBF(29),SECS30),(ADCTBF(28),SECS)
      C--   !CONTAINES THE ADISR 30 SEC. TIMER.
0061         EQUIVALENCE (ADCTBF(30),LFLSHR)
      C--   !CONTAINES THE BITS OF THE LIGHTS TO FLASH.
      C--
      C--   INITILALIZE PARAMETERS
      C--
0062         DATA RST, EX, REC/'* REST *', 'EXERCISE', 'RECOVERY'/
0063         DATA M/'M'/,F/'F'/
      C--   !SET VAIABLES M AND F EQUAL TO THEIR ACSII VALUES.
0064         DATA DROUTB/"167772/
      C--   !DATA OUT BUFFER FOR THE PARALLEL INTERFACE CARD.
0065         DATA DRINB/"167774/
      C--   !DATA IN BUFFER FOR THE PARALLEL INTERFACE CARD.
0066         DATA PCSR/"172540/
      C--   !OCTAL ADDRESS OF THE PROGRAMABLE CLOCK(KW11-P)
      C     !STATUS REGISTER.
0067         DATA PCCB/"172542/
      C--   !OCTAL ADDRESS OF THE PROGRAMABLE CLOCK
      C     !COUNTER BUFFER.
0068         DATA PCC/"144/
      C--   !SET CLOCK TO GET 100 INTERRUPS/SEC.
0069         DATA PCS/"113/
```

```
      C--   !OCTAL VALUE TO BE LOADED INTO THE CLOCK STATUS
      C     !REGISTER. BITS 0,1,3, AND 6 ARE TURNED ON.
      C     !   BIT 6 - ALLOWS DONE TO CAUSE AN INTERRUPT.
      C     !   BIT 3 - SELECTS REPEATED INTERRUPT MODE.
      C     !   BIT 1 - SET CLOCK RATE TO 10KHZ.
      C     !   BIT 0 - STARTS THE CLOCK.
0070          DATA OUT/'OUT '/
0071          SVTHRS=IFIX((0.2-SLPYCP(2))/SLPYCP(1))
0072          HRTHRS=IFIX((35.-SLPYCP(10))/SLPYCP(9))
0073          FVC=0
      C--   !INITIALIZE FVC FLAG.
0074          PFT=0
      C--   !INITILIZE THE PFT FLAG.
      D       TYPE 5
      D5      FORMAT(' BEFORE QUEUE')
0075          IF(IQSET(5).NE.0) STOP 'QUEUE ALLLCCATION FAILURE'
      C--
0077  100     CONTINUE
      C--
      C--   SET UP THE REST PART
      C--
0078          SECTN=RST
0079          CALL LIGHT("2)
0080          LITES="26
0081          ICT=61
0082          LFLSHR="2      !SET REST LIGHT TO FLASH
0083          IDATA(64)=0
0084          IDATA(68)=0
0085          KCNT=0   !INITILIZE RECORD COUNTER
0086          ASSIGN 130 TO IRETRN                    !ASSIGN RETURN LABEL FOR START
0087          GO TO 520                     !GO TO INITIALIZE & START ROUTINE.
0088  130     CONTINUE
0089          IF(EXHCNT.EQ.0)GO TO 135
      D       TYPE 3,GASDSP,EXHCNT
      D3      FORMAT(2X,'GASDSP=',I3,'EXHCNT=',I3)
0091          CALL BREATH
      D       TYPE 4,VO2,VCO2,VOL,IBRTH
      D4      FORMAT(2X,'VO2=',F6.1,'VCO2=',F6.1,'VOL=',F6.1,'IBRTH=',I3)
0092  135     CONTINUE
0093          IF(SECS30.EQ.0)GO TO 145
0095          WL=0.
0096            AS=0.
0097          CALL TRSBUF
0098          KCNT=KCNT+1
0099          CALL RPT30
0100  145     CONTINUE
0101          IRES=IPEEK(DRINB)                       !CHECK CONTROL PANEL PUSHBUTTON
0102          IF(IRES.EQ."2)GO TO 100                 !RESTART REST
0104          IF(IRES.EQ."20)GO TO 160     !PAUSE
0106          IF(IRES.EQ."4)GO TO 200                 !START EXERCISE
0108          IF(KCNT.GE.20)GO TO 200                 !MAX RECORDS REACHED
0110          GO TO 130                     !GET NEXT 30.SEC RECORD
0111  160     ASSIGN 145 TO IWAIT1                    !ASSIGN RETURN LABEL FOR PAUSE
0112          GO TO 562                     !GO TO PAUSE ROUTINE
```

```
0113  200   CONTINUE
0114        CALL IPOKE(PCSR,0)            !STOP CLOCK.
0115        IDATA(64)=(KCNT+1)/2
0116        CALL DATA (OUT)
0117        IF(IRES.EQ.4) GO TO 220
      C     WAIT UNTIL EX BUTTON IS PUSHED
0119  205   IF(IPEEK(DRINB).NE."4) GO TO 205
0121  207   IF(IPEEK(DRINB).NE.0) GO TO 207                 !WAIT TILL EX RELEASED
      C--
      C--  SET UP EXERCIZE PART
0123  220   CALL IPOKE(PCSR,0)
0124        SECTN=EX
0125        CALL LIGHT("4)
0126        LITES="34
0127        LFLSHR="4
0128        ICT=65
      C--
0129        IDATA(68)=0
0130        KCNT=0                        !RECORD COUNT
0131        MREC=2*(60-IDATA(64))
0132        ILAST=0
0133        ASSIGN 230 TO IRETRN          !ASSIGN RETURN LABEL FOR START ROUTINE
0134        GO TO 520           !GO TO INITIALIZE & START ROUTINE
0135  230   CONTINUE
0136        IF(SECS.EQ.ILAST)GO TO 233
0138        CALL WRKADJ
0139        ILAST=SECS
0140  233   CONTINUE
0141        IF(EXHCNT.EQ.0)GO TO 235
0143        CALL BREATH
0144  235   CONTINUE
0145        IF(SECS30.EQ.0)GO TO 245
0147        CALL TRSBUF
0148        KCNT=KCNT+1
0149        CALL RPT30
0150  245   CONTINUE
0151        IRES=IPEEK(DRINB)             !CHECK USER RESPONCE
0152        IF(IRES.EQ."4)GO TO 220              !RESTART EXERCISE
0154        IF(IRES.EQ."20)GO TO 260             !PAUSE
0156        IF(IRES.EQ."10)GO TO 300      !START RECOVERY
0158        IF(KCNT.GE.MREC)GO TO 300     !EXERCISE DONE, MAX # OF RECORDS
0160        GO TO 230                     !GET NEXT RECORD
0161  260   ASSIGN 245 TO IWAIT1          !ASSIGN RETURN LABEL FOR PAUSE ROUTINE
0162        GO TO 562           !GO TO PAUSE ROUTINE
0163  300   CONTINUE
0164        CALL IPOKE(PCSR,0)
0165        IDATA(68)=(KCNT+1)/2
0166        CALL DATA(OUT)
0167        IF(KCNT.GE.MREC)GO TO 900  !MAX REACHED, GO TO END
      C--
      C--  SET UP RECOVERY PART
      C--
0169  310   CONTINUE
0170        CALL IPOKE(PCSR,0)
```

```
0171          MREC=2*(60-IDATA(64)-IDATA(68))
0172          KCNT=0
0173          SECTN=REC
0174          CALL LIGHT("10)
0175          LITES="70
0176          LFLSHR="10
0177          ICT=69
0178          ASSIGN 330 TO IRETRN          !ASSIGN RETURN LABEL FOR START ROUTINE
0179          GO TO 520              !GO TO INITIALIZE & START ROUTINE.
0180    330   CONTINUE
0181          IF(EXHCNT.EQ.0)GO TO 335
0183          CALL BREATH
0184    335   CONTINUE
0185          IF(SECS30.EQ.0)GO TO 345
0187          WL=0.
0188          AS=0.
0189          CALL IRSBUF
0190          KCNT=KCNT+1
0191          CALL RPT30
0192    345   CONTINUE
0193          IRES=IPEEK(DRINB)              !GET USER RESPONCE
0194          IF(IRES.EQ."10)GO TO 310               !RESTART RECOVERY
0196          IF(IRES.EQ."20)GO TO 360               !PAUSE
0198          IF(IRES.EQ."40)GO TO 400      !RECOVERY DONE (END OF TEST)
0200          IF(KCNT.GE.MREC) GO TO 400   !MAX RECORDS
0202          GO TO 330    !GET NEXT RECORD
0203    360   ASSIGN 345 TO IWAIT1                   !ASSIGN RETURN LABEL FOR PAUSE
0204          GO TO 562                      !GO TO PAUSE ROUTINE
0205    400   CONTINUE
0206          CALL IPOKE(PCSR,0)             !TURN OFF CLOCK
0207    900   IDATA(72)=(KCNT+1)/2
        C     TYPE  1
        C1    FORMAT(' BEFORE LIGHT')
0208          CALL LIGHT(0)                          !TURN OFF ALL LIGHTS
0209          RETURN
        C
        C     ROUTIINE TO INITIALIZE & START EACH SECTION OF EXERCISE TEST
0210    520   IPAUSE=-1                      !INITIALIZE PAUSE FLAG
0211          CALL TIMRD(IHR,IMIN,ISEC)
0212          IDATA(ICT)=IHR
0213          IDATA(ICT+1)=IMIN
0214          IDATA(ICT+2)=ISEC
0215    522   CONTINUE
0216          ISEC=IDATA(63)
0217          IF(IPAUSE.EQ.-1 .AND. SECTN.EQ.REC) GO TO 524
0219    523   CALL INITI
        C     TYPE 9,EXHCNT
        C9    FORMAT(2X,'EXHCNT=',I3)
0220          BPTIME=550 - (10*ISEC)
0221          IF(BPTIME.LT.0) BPTIME=BPTIME + 600
0223          VO2=0.0
0224          VCO2=0.0
0225          VOL=0.0
0226          IBRTH=0
```

```
        D       TYPE 10, VO2,VCC2,VCL,IBRTH
        D10     FORMAT(2X,3F7.1,I3)
0227    524     CALL IPOKE(PCCR,PCC)        !SET PROGRAMMABLE CLOCK TO 100 HZ
0228            CALL IPOKE(PCSR,PCS)        !SET STATUS IN CLOCK (START)
0229            CALL LIGHT(LITES)   !TURN ON APPROPRIATE PANEL LIGHTS
0230            IF(IPAUSE.NE.-1) GO TO 526
0232            IF(SECTN.EQ.RST) CALL RPTHDR
0234            TYPE 525,SECTN,IHR,IMIN,ISEC
0235    525     FORMAT(/5X,'* * * ',A8,' * * *          REAL TIME: ',I2,':',
               @I2,':',I2/)
0236    526     IF(IPAUSE.EQ.0) TYPE 527,JHR,JMIN,JSEC
0238    527     FORMAT(/' CONTINUE AT ', I2,':',I2,':',I2/)
0239    528     IF(IPEEK(DRINB).NE.0) GO TO 528      !WAIT TILL SWITCHES ARE RELEAS
0241            GO TO IRETRN                         !RETURN
        C
        C       PAUSE ROUTINE
        C
0242    562     IF(IPAUSE.EQ.1)GO TO 570    !ENDING PAUSE PERIOD?
0244            CALL IPOKE(PCSR,0)          !(START OF PAUSE) STOP CLOCK
0245            IPAUSE=1
0246            TYPE 564
0247    564     FORMAT(/' * PAUSE *')
0248            CALL LIGHT (LITES)          !TURN ON APPROPRIATE LIGHTS
0249    565     IF(IPEEK(DRINB).NE.0) GO TO 565 !WAIT TILL SWITCHES ARE RELEASED
0251            GO TO IWAIT1                         !GO WAIT FOR ANOTHER BUTTON PUS
0252    570     IPAUSE=0                    !CAME HERE IF ENDING PAUSE
0253            CALL TIMRD(JHR,JMIN,JSEC)
0254            ISEC=JSEC
0255            GO TO 523                   !END OF PAUSE ROUTINE
0256            END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

RETI     000014  LOGICAL*1 ARRAY (8)
RETIX    000024  LOGICAL*1 ARRAY (8)
OUT      000104  INTEGER*4 VARIABLE
M        000064  INTEGER*2 VARIABLE
F        000066  INTEGER*2 VARIABLE
PCSR     000074  INTEGER*2 VARIABLE
PCCB     000076  INTEGER*2 VARIABLE
PCS      000102  INTEGER*2 VARIABLE
PCC      000100  INTEGER*2 VARIABLE
OLD      000314  INTEGER*2 VARIABLE
DROUTB   000070  INTEGER*2 VARIABLE
DRINB    000072  INTEGER*2 VARIABLE
SECTN    000316  REAL*8    VARIABLE
RST      000034  REAL*8    VARIABLE
EX       000044  REAL*8    VARIABLE
REC      000054  REAL*8    VARIABLE
IFIX     000000  INTEGER*2 PROCEDURE
IQSET    000000  INTEGER*2 PROCEDURE
LIGHT    000000  INTEGER*2 PROCEDURE
LITES    000326  INTEGER*2 VARIABLE
ICT      000330  INTEGER*2 VARIABLE
IRETRN   000332  INTEGER*2 VARIABLE
BREATH   000000  REAL*4    PROCEDURE
TRSBUF   000000  REAL*4    PROCEDURE
RPT30    000000  REAL*4    PROCEDURE
IRES     000334  INTEGER*2 VARIABLE
IPEEK    000000  INTEGER*2 PROCEDURE
IWAIT1   000336  INTEGER*2 VARIABLE
IPOKE    000000  INTEGER*2 PROCEDURE
DATA     000000  REAL*4    PROCEDURE
MREC     000340  INTEGER*2 VARIABLE
ILAST    000342  INTEGER*2 VARIABLE
WRKADJ   000000  REAL*4    PROCEDURE
IPAUSE   000344  INTEGER*2 VARIABLE
TIMRD    000000  REAL*4    PROCEDURE
IHR      000346  INTEGER*2 VARIABLE
IMIN     000350  INTEGER*2 VARIABLE
ISEC     000352  INTEGER*2 VARIABLE
INITI    000000  INTEGER*2 PROCEDURE
RPTHDR   000000  REAL*4    PROCEDURE
JHR      000354  INTEGER*2 VARIABLE
JMIN     000356  INTEGER*2 VARIABLE
JSEC     000360  INTEGER*2 VARIABLE


COMMON BLOCK /GAS30/    LENGTH 000016

VO2      000000  REAL*4    VARIABLE
VCO2     000004  REAL*4    VARIABLE
VOL      000010  REAL*4    VARIABLE
IBRTH    000014  INTEGER*2 VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

COMMON BLOCK /TTMCNT/    LENGTH 000014

TIMBUF   000000   INTEGER*2 ARRAY (6)
EXHCNT   000012   INTEGER*2 VARIABLE

COMMON BLOCK /PDATA/     LENGTH 006000

IDATA    000000   INTEGER*2 ARRAY (1536)
SLPYCP   000250   REAL*4    ARRAY (18)
SSNO     000004   INTEGER*2 ARRAY (3)
UNIQNO   000014   INTEGER*2 ARRAY (2)
RETNO    000020   INTEGER*2 VARIABLE
CPID     000022   INTEGER*2 VARIABLE
DATE     000116   INTEGER*2 ARRAY (3)
NAME     000062   INTEGER*2 ARRAY (14)
SEX      000002   INTEGER*2 VARIABLE
AGE      000060   INTEGER*2 VARIABLE
HIGHT    000142   REAL*4    VARIABLE
WT       000146   REAL*4    VARIABLE
KCNT     000216   INTEGER*2 VARIABLE

COMMON BLOCK /GASCOM/    LENGTH 007640

GASBUF   000000   INTEGER*2 ARRAY (2000)

COMMON BLOCK /ADCOM1/    LENGTH 000104

ADCTBF   000000   INTEGER*2 ARRAY (34)
BPTIME   000044   INTEGER*2 VARIABLE
BPCNT1   000052   INTEGER*2 VARIABLE
BPCNT    000064   INTEGER*2 VARIABLE
GASDSP   000000   INTEGER*2 VARIABLE
EXHALE   000002   INTEGER*2 VARIABLE
SVTHRS   000004   INTEGER*2 VARIABLE
FVC      000006   INTEGER*2 VARIABLE
PUSHES   000014   REAL*4    VARIABLE
PFT      000016   INTEGER*2 VARIABLE
HRTHRS   000024   INTEGER*2 VARIABLE
WL       000036   REAL*4    VARIABLE
AS       000046   REAL*4    VARIABLE
SECS30   000070   INTEGER*2 VARIABLE
SECS     000066   INTEGER*2 VARIABLE
LFLSHR   000072   INTEGER*2 VARIABLE
```

```
C
C----------------------------------------------------------
      SUBROUTINE WRKADJ
C----------------------------------------------------------
C     AUTHOR:  CHARLES MANN     CONSULTANTS: PAUL SCHACHTER
C     DATE:    JULY 26, 1978                 STAN FINK
C     PURPOSE: TO ADJUST THE WORK LOAD ON THE BICYCLE SO THAT A
C     SUBJECT'S HEART RATE MATCHES A GIVEN TARGET VALUE
C     AT A GIVEN TARGET TIME.
C     SUBROUTINE: SNATCH  (ASSEMBLY LANGUAGE)
C-----------------------------------------------------------------------
      INTEGER DAC1,DAC2,OUTPUT,IADDAT(34),ICALIB(1536)
      INTEGER*4 IHRAD,IWLAD
      REAL HRARY(30),WLARY(30)
      INTEGER IHRD(4)
      COMMON /ADCOM1/ IADDAT
      COMMON /PDATA/ ICALIB
      EQUIVALENCE (IADDAT(28),ICOUNT)     !ONE SECOND TIMER
      EQUIVALENCE (ICALIB(101),HRSLOP)
      EQUIVALENCE (ICALIB(103),HRINT)
      EQUIVALENCE (ICALIB(105),WLSLOP)
      EQUIVALENCE (ICALIB(107),WLINT)
      EQUIVALENCE (ICALIB(15),IHRD(1))    !TARGET HEART RATES
      DATA WL/0./
      DATA CONST1/.25/,CONST2/0./
      DATA DAC1/"176760/,DAC2/"176762/
      DATA ISTAND/0/,IHRCPT/1/,HRS/80./
      DATA HRARY/30*80./
C  CHECK FOR TREADMILL--THIS ROUTINE WORKS ONLY FOR THE BIKE
      IF (ICALIB(12).NE.1) RETURN
C  SHIFT HR INTO ARRAY
      DO 10 I=1,29
      WLARY(I)=WLARY(I+1)
      HRARY(I)=HRARY(I+1)
 10   CONTINUE
C  GET THE HR & WL DATA STORED BY THE ADISR
      CALL SNATCH(IHRAD,IWLAD,IHRCNT,IWLCNT)
C  CHANGE A/D HEART RATE INTO BPM
      IF((IHRCNT.EQ.0).OR.(IWLCNT.EQ.0)) GO TO 13
      HRARY(30)=HRSLOP*(AJFLT(IHRAD)/FLOAT(IHRCNT))+HRINT
      WLARY(30)=WLSLOP*(AJFLT(IWLAD)/FLOAT(IWLCNT))+WLINT
      ACTWL=WLARY(30)
      ACTHR=HRARY(30)
 13   IF (MOD(ICOUNT,5).NE.0) GO TO 999
      IF(ICOUNT.EQ.5) GO TO 110
      IF(ICOUNT.EQ.240) GO TO 120
      IF(ICOUNT.EQ.480) GO TO 120
      IF(ICOUNT.EQ.720) GO TO 120
      IF((240.LT.ICOUNT).AND.(ICOUNT.LT.300)) GO TO 118
      IF((480.LT.ICOUNT).AND.(ICOUNT.LT.540)) GO TO 118
      IF((720.LT.ICOUNT).AND.(ICOUNT.LT.780)) GO TO 118
      IF (ICOUNT.GT.960) GO TO 103
C  STANDARD FLOATING WL ROUTINE
 80   HRSOLD=HRS
```

```
      C  CALCULATE HRS(SMOOTH HR AVG.)
0051        DO 60 I=26,30
0052  60    HRS=HRS+HRARY(I)
0053        HRS=HRS/6.
      C  CHECK TO SEE IF HR IS CLOSE ENOUGH
0054        IF (ABS(FLOAT(IHRC(IHRCPT))-HRS).LE.1.0) GO TO 999
      C  DON'T LET HR CHANGE TOO QUICKLY
0056        IF (ABS(HRS-HRSOLD).GT.5.0) GO TO 999
      C  CALCULATE NEW WL
0058        WLOLD=WL                             !SAVE OLD WL
0059        WL=WLOLD+CONST1*(FLOAT(IHRC(IHRCPT))-HRS)+CONST2*((HRS-HRSOLD)
          &     /ABS(FLOAT(IHRC(IHRCPT))-HRS))
0060        GO TO 140
      C  DETERMINE THE RESTING HEART RATE
0061  110   WLFREE=0.
      C  CALCULATE THE AVERAGE RESTING HR FROM PRE-EXERCISE RECORDS
0062        IREST=ICALIB(64)
0063        IR64=MOD(IREST,2)
0064        IRECNO=(IREST/2)+IR64
0065        IF (IRECNO.GT.0) GO TO 111
0067        HRFREE=65.
0068        GO TO 113
0069  111   HRFREE=0.
0070        DO 112 I=1,IRECNO
0071  112   HRFREE=HRFREE+ICALIB(258+(I-1)*20)
0072        HRFREE=HRFREE/IRECNO
0073        IF (HRFREE.LT.45.) HRFREE=45.
0075        IF (HRFREE.GT.70.) HRFREE=70.
      C  SET HRD
0077  113   IHRDPT=1
      C  INITIALIZE WL
0078        WL=0.
      C  GO TO STANDARD WL ROUTINE
0079        GO TO 80
      C  END OF EXERCISE--DECREASE WL TO 10% OF END IN 90 SEC
0080  103   WL=WL*.85
0081        GO TO 140
0082  117   ISTAND=1
0083  118   IF (ISTAND.EQ.1) GO TO 80
0085        GO TO 130
      C  SET NEW DESIRED HR
0086  120   IHRDPT=IHRDPT+1
      C  INITIALIZE FLAG TO HOLD A CONSTANT WL
0087        ISTAND=0
      C  SAVE OLD WL & HR
0088        WLPAST=WLFREE
0089        HRPAST=HRFREE
      C  CALCULATE THE CONSTANT WL VALUE
0090        WLFREE=0.
0091        HRFREE=0.
0092        DO 125 I=1,30
0093         WLFREE=WLFREE+WLARY(I)
0094  125   HRFREE=HRFREE+HRARY(I)
0095        HRFREE=HRFREE/30.
```

```
0096            WLFREE=WLFREE/30.
0097            SLWL=(WLFREE-WLPAST)/(HRFREE-HRPAST)
0098            WLCNST=WLFREE
0099            IF(SLWL.LE.0) GO TO 117
        C  IF TARGET HR IS LESS THAN THE CURRENT AVE. HR OR CLOSE TO IT,
        C  GO TO STANDARD CORRECTOR WL RATHER THAN A CONSTANT WL.
0101            IF ((IFIX(HRFREE)-IHRD(IHRDPT)).GT.-5) GO TO 117
0103            WLCNST=WLFREE+SLWL*(FLOAT(IHRD(IHRDPT))-HRFREE)
        C  KEEP WORK LOAD CONSTANT
0104    130     WL=WLCNST
        C     OUTPUT THE WL VALUE TO THE BICYCLE
        C  CHECK FOR MAX WORKLOAD
0105    140     IF (WL.GT.300.) WL=300.
        C  CHECK FOR MIN WORKLOAD
0107            IF (WL.LT.0) WL=0.
0109            OUTPUT=IFIX(6.144*WL)
0110            CALL IPOKE(DAC1,OUTPUT)
0111            CALL IPOKE(DAC2,OUTPUT)                  !BACKUP CHANNEL
0112    999     CONTINUE
        CC      IF (MOD(ICOUNT,15).NE.0) RETURN
        CC      TYPE 825,ICOUNT,ACTHR,HRS,IHRD(IHRDPT),WL,ACTWL,WLFREE,HRFREE
        CC825 FORMAT(' TIME=',I4,3X,'IHR=',F4.0,3X'HRS=',F4.0,3X,'HRD=',I4,3X,
        CC      *    'CAL WL=',F4.0,3X,'ACT WL=',F4.0,3X
        CC      *    ,'WLF=',F4.0,3X,'HRF=',F4.0)
0113            RETURN
0114            END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

HRARY    000014   REAL*4     ARRAY (30)
WLARY    000204   REAL*4     ARRAY (30)
DAC1     000410   INTEGER*2 VARIABLE
DAC2     000412   INTEGER*2 VARIABLE
OUTPUT   000440   INTEGER*2 VARIABLE
IHRAD    000442   INTEGER*4 VARIABLE
IWLAD    000446   INTEGER*4 VARIABLE
WL       000374   REAL*4     VARIABLE
CONST1   000400   REAL*4     VARIABLE
CONST2   000404   REAL*4     VARIABLE
ISTAND   000414   INTEGER*2 VARIABLE
IHRDPT   000416   INTEGER*2 VARIABLE
HRS      000420   REAL*4     VARIABLE
I        000452   INTEGER*2 VARIABLE
SNATCH   000000   REAL*4     PROCEDURE
IHRCNT   000454   INTEGER*2 VARIABLE
IWLCNT   000456   INTEGER*2 VARIABLE
AJFLT    000000   REAL*4     PROCEDURE
FLOAT    000000   REAL*4     PROCEDURE
ACTWL    000460   REAL*4     VARIABLE
ACTHR    000464   REAL*4     VARIABLE
MOD      000000   INTEGER*2 PROCEDURE
HRSOLD   000470   REAL*4     VARIABLE
ABS      000000   REAL*4     PROCEDURE
WLOLD    000474   REAL*4     VARIABLE
WLFREE   000500   REAL*4     VARIABLE
IREST    000504   INTEGER*2 VARIABLE
IR64     000506   INTEGER*2 VARIABLE
IRECNO   000510   INTEGER*2 VARIABLE
HRFREE   000512   REAL*4     VARIABLE
WLPAST   000516   REAL*4     VARIABLE
HRPAST   000522   REAL*4     VARIABLE
SLWL     000526   REAL*4     VARIABLE
WLCNST   000532   REAL*4     VARIABLE
IFIX     000000   INTEGER*2 PROCEDURE
IPOKE    000000   INTEGER*2 PROCEDURE


COMMON BLOCK /ADCOM1/   LENGTH 000104

IADDAT   000000   INTEGER*2 ARRAY (34)
ICOUNT   000066   INTEGER*2 VARIABLE

COMMON BLOCK /PDATA/    LENGTH 006000

ICALIB   000000   INTEGER*2 ARRAY (1536)
HRSLOP   000310   REAL*4     VARIABLE
HRINT    000314   REAL*4     VARIABLE
WLSLOP   000320   REAL*4     VARIABLE
WLINT    000324   REAL*4     VARIABLE
IHRD     000034   INTEGER*2 ARRAY (4)
```

```
SNATCH  RT-11 MACRO VM02-12   25-FEB-80 15:07:16 PAGE 1

             ;------------------------------------------------------------
                     .TITLE  SNATCH
             ;------------------------------------------------------------
             ;       AUTHOR:  CHARLES MANN
             ;       DATE:    SEPTEMBER 21, 1978
             ;       PURPOSE: TO GET THE HR & WL DATA STORED IN COMMON BY THE INTERUPT
             ;                SERVICE ROUTINE AND TO INITIALIZE THESE LOCATIONS BACK TO
             ;                ZERO.
             ;............................................................
             ;
                     .GLOBL  SNATCH
                     .MCALL  .REGDEF,..V2..
00000                ..V2..
00000                .REGDEF
00000 106427 SNATCH: MTPS    #340            ;INHIBIT INTERRUPTS
      000340
00004 016501         MOV     2(R5),R1        ;STORE ADDRESS OF IHRAD INTO R1
      000002
00010 016721         MOV     CHR,(R1)+       ;GET LOWER ORDER WORD OF HR SUM
      000026'
00014 016711         MOV     CHR1,(R1)       ;GET HIGHER ORDER WORD OF HR SUM
      000030'
00020 016501         MOV     4(R5),R1        ;STORE ADDRESS OF IWLAD INTO R1
      000004
00024 016721         MOV     CWL,(R1)+       ;GET LOWER ORDER WORD OF WL SUM
      000076'
00030 016711         MOV     CWL1,(R1)       ;GET HIGHER ORDER WORD OF WL SUM
      000100'
00034 016775         MOV     CHRCNT,@6(R5)   ;GET NO. OF SAMPLES FOR HR
      000020'
      000006
00042 016775         MOV     CWLCNT,@10(R5)  ;GET NO. OF SAMPLES FOR WL
      000102'
      000010
00050 005067         CLR     CHR             ;ZERO THE WORDS IN PREPARATION
      000026'
00054 005067         CLR     CHR1            ;       FOR MORE DATA
      000030'
00060 005067         CLR     CWL
      000076'
00064 005067         CLR     CWL1
      000100'
00070 005067         CLR     CHRCNT
      000020'
00074 005067         CLR     CWLCNT
      000102'
00100 106427         MTPS    #0              ;ALLOW ADISR TO COLLECT MORE DATA
      000000
00104 000207         RTS     PC              ;RETURN TO WRKADJ ROUTINE
      000000'        .CSECT  ADCOM1          ;COMMON AREA WHERE ADISR STORES A/D DATA
                     .BLKW   8.
00020 000000 CHRCNT: .WORD
                     .BLKW   2
00026 000000 CHR:    .WORD
00030 000000 CHR1:   .WORD
                     .BLKW   18.
00076 000000 CWL:    .WORD
```

```
SNATCH  RT-11 MACRO VM02-12   29-FEB-80 15:07:10 PAGE 1+

40 00100 000000 CHL1:   .WORD
41 00102 000000 CHLCNT: .WORD
42        000001'       .END
```

```
SNATCH  RT-11 MACRO VM02-12   29-FEB-80 15:07:10 PAGE 1+
SYMBOL TABLE

CHR      000026R    002  CHRCNT  000020R    002  CHR1     000030R    002
CHL      000076R    002  CHLCNT  000102R    002  CHL1     000100R    002
PC     =%000007          R0    =%000000          R1     =%000001
R2     =%000002          R3    =%000003          R4     =%000004
R5     =%000005          SNATCH  000000RG        SP     =%000006
...V2 = 000001
. ABS.  000000      000
        000106      001
ADCOM1  000104      002
ERRORS DETECTED: 0
FREE CORE: 17884. WORDS

,LP:=SNATCH
```

```
TRSBUF
~~.MAIN.~~  RT-11 MACRO VM02-12   29-FEB-80 15:10:48 PAGE 1

 1                                  .GLOBL  TRSBUF
 2                                  .MCALL  .REGDEF,..V2..
 3 000000                           ..V2..
 4 000000                           .REGDEF
 5 000000 106427  TRSBUF:           MTPS    #200
          000200
 6 000004 010146                    MOV     R1,-(SP)
 7 000006 010246                    MOV     R2,-(SP)
 8 000010 010346                    MOV     R3,-(SP)
 9 000012 012701                    MOV     #AHRCNT,R1
          000022'
10 00016  012702                    MOV     #AHRC,R2
          000000'
11 00022  012703                    MOV     #20.,R3
          000024
12 00026  012122  LOOP:             MOV     (R1)+,(R2)+
13 00030  005303                    DEC     R3
14 00032  003375                    BGT     LOOP
15 00034  012701                    MOV     #V02,R1
          000000'
16 00040  012702                    MOV     #V02C,R2
          000000'
17 00044  012703                    MOV     #7.,R3
          000007
18 00050  012122  LOOP0:            MOV     (R1)+,(R2)+
19 00052  005303                    DEC     R3
20 00054  003375                    BGT     LOOP0
21 00056  012701                    MOV     #AHRCNT,R1
          000022'
22 00062  012703                    MOV     #20.,R3
          000024
23 00066  005021  LOOP1:            CLR     (R1)+
24 00070  005303                    DEC     R3
25 00072  003375                    BGT     LOOP1
26 00074  016767                    MOV     AHRTC,AHRT
          000002'
          000024'
27 00102  016767                    MOV     CHRC,CHR
          000004'
          000026'
28 00110  016767                    MOV     CHR1C,CHR1
          000006'
          000030'
29 00116  016767                    MOV     BPTC,BPTIME
          000022'
          000044'
30 00124  016767                    MOV     BPCT1C,BPCNT1
          000030'
          000052'
31 00132  012767                    MOV     #10.,BPCNT
          000012
          000064'
32 00140  016767                    MOV     SECSC,SECS
          000044'
          000066'
33 00146  012703                    MOV     #7.,R3
          000007
```

```
00152 012701            MCV     #V02,R1
      000000'
00156 005021 LOOP2:     CLR     (R1)+
00160 005303            DEC     R3
00162 003375            BGT     LOOP2
00164 012603            MCV     (SP)+,R3
00166 012602            MCV     (SP)+,R2
00170 012601            MCV     (SP)+,R1
00172 106427            MTPS    #0
      000000
00176 000207            RTS     PC

      000000'           .CSECT  ADCCM1
                        .BLKW   9.
00022 000000 AHRCNT:    .WCRD   0
00024 000000 AHRT:      .WCRD   0
00026 000000 CHR:       .WCRD   0
00030 000000 CHR1:      .WCRD   0
                        .BLKW   5.
00044 001046 BPTIME:    .WCRD   550.
                        .BLKW   2.
00052 000000 BPCNT1:    .WCRD   0.
                        .BLKW   4.
00064 000012 BPCNT:     .WCRD   10.
00066 000000 SECS:      .WORD   0
                        .BLKW   6
      000000'           .CSECT  GAS30
00000        V02:       .BLKW   7.
      000000'           .CSECT GAS30C
00000        V02C:      .BLKW   7.
      000000'           .CSECT  ADCCM2
00000 000000 AHRC:      .WCRD   0.
00002 000000 AHRTC:     .WORD   0
00004 000000 CHRC:      .WCRD   0
00006 000000 CHR1C:     .WCRD   0
                        .BLKW   5.
00022 000000 BPTC:      .WCRD   0
                        .BLKW   2.
00030 000000 BPCT1C:    .WCRD   0
                        .BLKW   5.
00044 000000 SECSC:     .WORD   0
                        .BLKW   1
      000001'           .ENC
```

TRSBUF
~~.MAIN.~~ RT-11 MACRO VM02-12   29-FEB-80 15:10:48 PAGE 1+

```
SYMBOL TABLE

AHRC      000000R    005  AHRCNT    000022R    002  AHRT      000024R    002
AHRTC     000002R    005  BPCNT     000064R    002  BPCNT1    000052R    002
BPCT1C    000030R    005  BPTC      000022R    005  BPTIME    000044R    002
CHR       000026R    002  CHRC      000004R    005  CHR1      000030R    002
CHR1C     000006R    005  LOOP      000026R         LOOP0     000050R
LOOP1     000066R         LOOP2     000156R         PC      =%000007
R0      =%000000          R1      =%000001          R2      =%000002
R3      =%000003          R4      =%000004          R5      =%000005
SECS      000066R    002  SECSC     000044R    005  SP      =%000006
TRSBUF    000000RG        VO2       000000R    003  VO2C      000000R    004
...V2 =   000001
. ABS.    000000     000
          000200     001
ADCOM1    000104     002
GAS30     000016     003
GAS30C    000016     004
ADCOM2    000050     005
ERRORS DETECTED: 0
FREE CORE: 17813. WORDS

,LP:=TRSBUF
```

```
0001          SUBROUTINE BREATH
        C-------------------------------------------------------------
        C
        C     VERSION: 2.0
        C     AUTHOR:  CHARLES MANN
        C     DATE:    SEPTEMBER 18, 1978
        C     MODIFIED BY:        WILLIAM G. CROSIER
        C     DATE:               10 SEP 79
        C     PURPOSE: TO DETERMINE THE OXYGEN CONSUMPTION AND THE CARBON
        C     DIOXIDE PRODUCTION IN EACH BREATH.
        C
        C.................................................................
        C
0002          INTEGER GASBUF(2000),CALIB(1536),TIME(6)
0003          REAL N2,N2SLP,N2INT,N2OUT,N2AMB
0004          COMMON /GASCOM/ GASBUF
0005          COMMON /PDATA/ CALIB
0006          COMMON /GAS30/ VO2,VCO2,VCL,IBRTH
0007          COMMON /TIMCNT/ TIME
0008          COMMON /ADCOM1/ IEOBPT,IDUMMY(33)   !END OF BREATH POINTER IN GASBUF
0009          EQUIVALENCE (CALIB(73),STPD)
0010          EQUIVALENCE (CALIB(75),BTPS)
0011          EQUIVALENCE (CALIB(77),O2AMB)
0012          EQUIVALENCE (CALIB(79),N2AMB)
0013          EQUIVALENCE (CALIB(81),CO2AMB)
0014          EQUIVALENCE (CALIB(85),VCLSLP)
0015          EQUIVALENCE (CALIB(87),VCLINT)
0016          EQUIVALENCE (CALIB(89),O2SLP)
0017          EQUIVALENCE (CALIB(91),O2INT)
0018          EQUIVALENCE (CALIB(93),N2SLP)
0019          EQUIVALENCE (CALIB(95),N2INT)
0020          EQUIVALENCE (CALIB(97),CO2SLP)
0021          EQUIVALENCE (CALIB(99),CO2INT)
0022          EQUIVALENCE (TIME(6),IEOBFG)        !END OF BREATH FLAG: COUNTS BREAT
        C  STORE SUBSCRIPT OF THE END OF THE BREATH SAMPLE
0023    5     IEOBP=IEOBPT
0024          IPOINT=IEOBP-3
0025          IF(IPOINT.LT.1)IPOINT=1
        CC    TYPE 1,(GASBUF(I),I=1,IEOBPT)
0027    1     FORMAT(/,1X,8(1X,I5))
        C  DECREMENT A/D BREATH COUNTER TO SHOW THAT THIS ONE IS BEING PROCESSED
0028    6     IEOBFG=IEOBFG-1
0029          IEOBF=IEOBFG
        C  INCREMENT THE BREATH COUNTER FOR THE 30 SEC REPORT
0030          IBRTH=IBRTH+1
        C  LOCATE MAXIMUM SPIROMETER VOLUME
0031          JPOINT=IPOINT-124
0032          IF(JPOINT.LT.1)JPOINT=1
0034          MAXVOL=GASBUF(JPOINT)
0035          IEND=JPOINT
0036          DO 10 I=JPOINT,IPOINT,4
0037            IF (GASBUF(I).LT.MAXVOL) GO TO 10
0039            MAXVOL=GASBUF(I)
0040            IEND=I              !SUBSCRIPT OF MAX VOLUME
```

```
10     CONTINUE
C
C  AVERAGE GAS CONCENTRATIONS AT LAST 5 POINTS OF BREATH TO REMOVE NOISE
       ISTART=IEND-16
       IF(ISTART.LT.1)ISTART=1
       IO2=0
       IN2=0
       ICO2=0
       DO 20 I=ISTART,IEND,4
         ICO2=ICO2+GASBUF(I+3)
         IO2=IO2+GASBUF(I+1)
         IN2=IN2+GASBUF(I+2)
20     CONTINUE
       IDIF=((IEND-ISTART)+4)/4
       IF(IDIF.LT.1)IDIF=1
       IO2=IO2/IDIF
       IN2=IN2/IDIF
       ICO2=ICO2/IDIF
C      AVERAGE LAST 3 SPIRO. VO. SAMPLES TO GET TIDAL VOLUME
       IF (IEND.GE.13) MAXVOL=(GASBUF(IEND-4)+GASBUF(IEND-8)
     &    +GASBUF(IEND-12))/3
C
C  USE CALIBRATION FACTORS TO CONVERT A/D COUNTS INTO PERCENTAGES
       O2=O2SLP*FLOAT(IO2)+O2INT
       N2=N2SLP*FLOAT(IN2)+N2INT
       CO2=CO2SLP*FLOAT(ICO2)+CO2INT
C  USE CALIBRATION FACTORS TO CONVERT A/D COUNTS INTO LITERS
       VOLOUT=VOLSLP*FLOAT(MAXVOL)+VOLINT
C  INCREMENT MINUTE VOLUME
       VOL=VOL+VOLOUT*BTPS
CC     TYPE 100,IBRTH,O2,N2,CO2,VOL
100    FORMAT(/,1X,'IBRTH  O2  N2  CO2  VOL ',I4,4(2X,F8.2))
C
C  NORMALIZE MASS SPEC. DATA TO CORRECT FOR DRIFT
       ALL=O2+N2+CO2
       O2OUT=O2/ALL
       N2OUT=N2/ALL
       CO2OUT=CO2/ALL
C
C  INCREMENT THE O2 CONSUMPTION AND CO2 PRODUCTION
       STPVOL = STPD * VOLOUT
       VO2 = VO2 + STPVOL * ((O2AMB/N2AMB)*N2OUT - O2OUT)
       VCO2 = VCO2 + STPVOL * (CO2OUT - (CO2AMB/N2AMB)*N2OUT)
C
C  CHECK TO SEE IF THERE IS ANOTHER BREATH TO DEAL WITH
       IF (IEOBFG.EQ.0) GO TO 900   ! NO MORE BREATHS, RETURN
       IF(IEOBF.EQ.0) GO TO 200     ! NO PRIOR BREATHS
C      FIND END OF PREVIOUS BREATH
       IREF=(0.3-VOLINT)/VOLSLP
       IPOINT=IEND
150    IPOINT=IPOINT-4
       IF (IPOINT.LT.41) GO TO 160 !BREATH .LT. 0.1 SEC. LONG
       IF(GASBUF(IPOINT).GT.IREF) GO TO 150
       GO TO 6
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:38:55                PAGE 003

0086  180    IEOBFG=IEOBFG-IEOBF           !COULDN'T FIND PREVIOUS EOB(S)
0087  200    IF (IEOBPT .GT. IEOBP) GO TO 5      ! A NEW BREATH IS FINISHED
        C  RESET POINTER TO THE BEGINNING OF THE GAS BUFFER
0089  900    IEOBPT=0
0090         RETURN
0091         END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET   ATTRIBUTES

N2       000102   REAL*4      VARIABLE
N2OUT    000106   REAL*4      VARIABLE
IEOBP    000112   INTEGER*2 VARIABLE
IPOINT   000114   INTEGER*2 VARIABLE
IEOBF    000116   INTEGER*2 VARIABLE
JPOINT   000120   INTEGER*2 VARIABLE
MAXVOL   000122   INTEGER*2 VARIABLE
IEND     000124   INTEGER*2 VARIABLE
I        000126   INTEGER*2 VARIABLE
ISTART   000130   INTEGER*2 VARIABLE
IO2      000132   INTEGER*2 VARIABLE
IN2      000134   INTEGER*2 VARIABLE
ICO2     000136   INTEGER*2 VARIABLE
IDIF     000140   INTEGER*2 VARIABLE
O2       000142   REAL*4      VARIABLE
FLOAT    000000   REAL*4      PROCEDURE
CO2      000146   REAL*4      VARIABLE
VOLOUT   000152   REAL*4      VARIABLE
ALL      000156   REAL*4      VARIABLE
O2OUT    000162   REAL*4      VARIABLE
CO2OUT   000166   REAL*4      VARIABLE
STPVOL   000172   REAL*4      VARIABLE
IREF     000176   INTEGER*2 VARIABLE


COMMON BLOCK /GASCOM/    LENGTH 007640

GASBUF   000000   INTEGER*2 ARRAY (2000)

COMMON BLOCK /PDATA/     LENGTH 006000

CALIB    000000   INTEGER*2 ARRAY (1536)
STPD     000220   REAL*4      VARIABLE
BTPS     000224   REAL*4      VARIABLE
O2AMB    000230   REAL*4      VARIABLE
N2AMB    000234   REAL*4      VARIABLE
CO2AMB   000240   REAL*4      VARIABLE
VOLSLP   000250   REAL*4      VARIABLE
VOLINT   000254   REAL*4      VARIABLE
O2SLP    000260   REAL*4      VARIABLE
O2INT    000264   REAL*4      VARIABLE
N2SLP    000270   REAL*4      VARIABLE
N2INT    000274   REAL*4      VARIABLE
CO2SLP   000300   REAL*4      VARIABLE
CO2INT   000304   REAL*4      VARIABLE

COMMON BLOCK /GAS30/     LENGTH 000016

VO2      000000   REAL*4      VARIABLE
VCO2     000004   REAL*4      VARIABLE
VOL      000010   REAL*4      VARIABLE
IBRTH    000014   INTEGER*2 VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

COMMON BLOCK /TIMCNT/   LENGTH 000014

TIME    000000  INTEGER*2 ARRAY (6)
IEOBFG  000012  INTEGER*2 VARIABLE

COMMON BLOCK /ADCOM1/   LENGTH 000104

IEOBPT  000000  INTEGER*2 VARIABLE
IDUMMY  000002  INTEGER*2 ARRAY (33)
```

```
          C--
          CC--  SUBROUTINE TO PRINT THE 30 SEC. REPORT
          C--
          C--  WRITTEN BY: ROY A. REED
          C--  DATE:       13-OCT-78
          C--
0001            SUBROUTINE RPT30
0002            INTEGER NUM,INUM,IBLK
0003            INTEGER DATBUF(20)
0004            INTEGER IBRTH,CWLCNT,AHRCNT,BPCNT,BPCNT1
0005            INTEGER*4 CSPD,AHR,CWL,CSBP,CDBP
0006            REAL DDATA(640)
0007            REAL VO2,VCO2,VCL,RDATA(24)
0008            REAL HR,WL,AS,SBP,DBP
0009            COMMON /GAS30C/VO2,VCO2,VCL,IBRTH
0010            COMMON /PDATA/IDATA(1536)
0011            COMMON /ADCOM2/DATBUF
0012            EQUIVALENCE (DATBUF(5),AHR),(DATBUF(7),CWL),
               %(DATBUF(9),CWLCNT),(DATBUF(11),CSPD),
               %(DATBUF(13),BPCNT1),(DATBUF(14),CSBP),
               %(DATBUF(16),CDBP),(DATBUF(18),BPCNT),
               %(DATBUF(1),AHRCNT)
0013            EQUIVALENCE (IDATA(257),DDATA(1))
0014            EQUIVALENCE (IDATA(72),NUM)
0015            EQUIVALENCE (IDATA(73),RDATA(1)),(IDATA(52),WT)
0016            DATA IBLK/2H  /
0017            DATA INUM/2H  /
0018            IF(IAJFLT(AHR,HR).EQ.-2)STOP
0020            IF(IAJFLT(CWL,WL).EQ.-2)STOP
0022            IF(IAJFLT(CSPD,AS).EQ.-2)STOP
0024            IF(IAJFLT(CSBP,SBP).EQ.-2)STOP
0026            IF(IAJFLT(CDBP,DBP).EQ.-2)STOP
0028            IF(HR.NE.0.)HR=(HR/FLOAT(AHRCNT))*RDATA(15)+RDATA(16)
0030            IF(WL.NE.0.)WL=(WL/FLOAT(CWLCNT))*RDATA(17)+RDATA(18)
0032            IF(AS.NE.0.)AS=(AS/FLOAT(CWLCNT))*RDATA(19)+RDATA(20)
0034            IF(SBP.NE.0.)SBP=(SBP/FLOAT(BPCNT))*RDATA(21)+RDATA(22)
0036            IF(DBP.NE.0.)DBP=(DBP/FLOAT(BPCNT))*RDATA(23)+RDATA(24)
0038            IF(NUM.EQ.1)TWL=0.
0040            IF(HR.LT.0.)HR=0.
0042            IF(WL.LT.0.)WL=0.
0044            IF(AS.LT.0.)AS=0.
0046            IF(SBP.LT.35.)SBP=0.
0048            IF(DBP.LT.35.)DBP=0.
0050            VO2LM=0.
0051            VCO2LM=0.
0052            VO2MKM=0.
0053            TVOL=0.
0054            RER=0.
0055            IRR=0
0056            RMVO2=0.
0057            RMVCO2=0.
0058            VO2HRR=0.
0059            IREM=MOD(NUM,2)
0060            NNUM=NUM/2
```

```
         IF(IREM.EQ.0)ENCODE(2,51,INUM) NNUM
 51      FORMAT(I2)
         VO2LM=VC2*2.
         VO2MKM=(VO2LM*1000.)/WT
         VCO2LM=VCO2*2.
         TVOL=VOL*2.
         IF(VO2LM.GT.0.)RER=VCO2LM/VC2LM
         IRR=IBRTH*2
         IF(VO2LM.GT.0.)RMVC2=TVOL/VC2LM
         IF(VCO2LM.GT.0)RMVCC2=TVCL/VCO2LM
         IF(HR.NE.0.)VC2HRB=(VO2LM*1000.)/HR
         IHR=IFIX(HR+.5)
         ISBP=IFIX(SBP+.5)
         IDBP=IFIX(DBP+.5)
         IF(IDATA(12).EG.2)GC TO 200
         TWL=TWL+(WL/2.)
         ITWL=IFIX(TWL+.5)
         IWL=IFIX(WL+.5)
         GO TO 300
 200     CONTINUE
         IWL=IFIX(WL*10.+.5)
           TWL=AS
           ITWL=IFIX(AS+.5)
 300     TYPE 400,INUM,IHR,ISBP,IDBP,WL,TWL,VO2LM,VO2MKM,VCO2LM,
        %TVOL,RER,IRR,RMVO2,RMVCO2,VC2HRB
 400     FORMAT(1X,A2,3(1X,I7),F8.1,F7.1,F8.2,F8.1,2X,
        %F8.2,F8.1,F7.2,I7,1X,3F9.1)
         IREST=ICATA(64)
         IEX=IDATA(68)
         NPT=IREST+IEX+(NUM-1)/2
         K=257+NPT*20
         KR=4+NPT*10
         KK=2-IREM
 CC      TYPE 450,IDATA(64),IDATA(68),NUM,NPT,K,KR,KK,IREM
 450     FORMAT(/,1X,' ICATA 64  ICATA 68  NUM  NPT ',4(1X,I5),
        %/,1X,' K  KR  KK  IREM ',4(1X,I5))
         IF(IREM.EQ.0)GC TC 550
         KKK=K+19
         DO 500 KZ=K,KKK
 500     IDATA(KZ)=0
 550     IF(NUM.NE.0)ICATA(K)=(NUM/2)+IREM
         IF(IHR.NE.0)ICATA(K+1)=(ICATA(K+1)+IHR)/KK
         IF(ISBP.NE.0)ICATA(K+2)=(ICATA(K+2)+ISBP)
         IF(IDBP.NE.0)ICATA(K+3)=(ICATA(K+3)+IDPP)
         IF(IWL.NE.C)ICATA(K+4)=(ICATA(K+4)+IWL)/KK
         IF(IRR.NE.0)ICATA(K+5)=(ICATA(K+5)+IRR)/KK
         IF(VO2LM.NE.0)CCATA(KR)=(CCATA(KR)+VC2LM)/FLOAT(KK)
         IF(VCO2LM.NE.0)CDATA(KR+1)=(DDATA(KR+1)+VCC2LM)/FLOAT(KK)
         IF(TVOL.NE.0)CCATA(KR+2)=(CCATA(KR+2)+TVOL)/FLOAT(KK)
         IF(AS.NE.0)DCATA(KR+3)=(CCATA(KR+3)+AS)/FLOAT(KK)
         IF(ITWL.NE.0.ANC.ICATA(12).EQ.1)ICATA(6)=ITWL
         INUM=IBLK
         RETURN
         END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

INUM     000016   INTEGER*2 VARIABLE
IBLK     000014   INTEGER*2 VARIABLE
HR       000206   REAL*4    VARIABLE
WL       000212   REAL*4    VARIABLE
AS       000216   REAL*4    VARIABLE
SBP      000222   REAL*4    VARIABLE
DBP      000226   REAL*4    VARIABLE
IAJFLT   000000   INTEGER*2 PROCEDURE
FLOAT    000000   REAL*4    PROCEDURE
TWL      000232   REAL*4    VARIABLE
VO2LM    000236   REAL*4    VARIABLE
VCO2LM   000242   REAL*4    VARIABLE
VO2MKM   000246   REAL*4    VARIABLE
TVOL     000252   REAL*4    VARIABLE
RER      000256   REAL*4    VARIABLE
IRR      000262   INTEGER*2 VARIABLE
RMVO2    000264   REAL*4    VARIABLE
RMVCO2   000270   REAL*4    VARIABLE
VO2HRR   000274   REAL*4    VARIABLE
IREM     000300   INTEGER*2 VARIABLE
MOD      000000   INTEGER*2 PROCEDURE
NNUM     000302   INTEGER*2 VARIABLE
IHR      000304   INTEGER*2 VARIABLE
IFIX     000000   INTEGER*2 PROCEDURE
ISBP     000306   INTEGER*2 VARIABLE
IDBP     000310   INTEGER*2 VARIABLE
ITWL     000312   INTEGER*2 VARIABLE
IWL      000314   INTEGER*2 VARIABLE
IREST    000316   INTEGER*2 VARIABLE
IEX      000320   INTEGER*2 VARIABLE
NPT      000322   INTEGER*2 VARIABLE
K        000324   INTEGER*2 VARIABLE
KR       000326   INTEGER*2 VARIABLE
KK       000330   INTEGER*2 VARIABLE
KKK      000332   INTEGER*2 VARIABLE
KZ       000334   INTEGER*2 VARIABLE


COMMON BLOCK /GAS30C/   LENGTH 000016

VO2      000000   REAL*4    VARIABLE
VCO2     000004   REAL*4    VARIABLE
VOL      000010   REAL*4    VARIABLE
IBRTH    000014   INTEGER*2 VARIABLE

COMMON BLOCK /PDATA/    LENGTH 006000

IDATA    000000   INTEGER*2 ARRAY (1536)
DDATA    001000   REAL*4    ARRAY (640)
NUM      000216   INTEGER*2 VARIABLE
RDATA    000220   REAL*4    ARRAY (24)
WT       000146   REAL*4    VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

COMMON BLOCK /ADCOM2/   LENGTH 000050

DATBUF   000000  INTEGER*2 ARRAY (20)
AHR      000010  INTEGER*4 VARIABLE
CWL      000014  INTEGER*4 VARIABLE
CWLCNT   000020  INTEGER*2 VARIABLE
CSPD     000024  INTEGER*4 VARIABLE
BPCNT1   000030  INTEGER*2 VARIABLE
CSBP     000032  INTEGER*4 VARIABLE
CDBP     000036  INTEGER*4 VARIABLE
BPCNT    000042  INTEGER*2 VARIABLE
AHRCNT   000000  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:46:30                 PAGE 001

          C-- SUBROUTINE RPTHDR, VERSION 2.0    29 AUG. 1979
          C-- PRINTS HEADING FOR 30-SECOND SUMMARY REPORT FOR PROCES
          C
0001             SUBROUTINE RPTHDR
0002             INTEGER EXSTRT,RCSTRT
0003             INTEGER AGE,UNIQ,TESTRT,WLSUM
0004             REAL WT,HT,AMTEMP,AMPRES
0005             COMMON/PDATA/IDUM1,ISEX,NMSSEC(3),WLSUM,
                @UNIQ(2),NMRTST,ICPIC,IMODE,ITSTYP,
                @IDUM3(2),IHR4,IHR8,IHR12,IHR16,IDUM4,IHRMAX,
                @IDUM5,IBTHDY(3),AGE,NAME(14),ITDATE(3),
                @IDUM6(7),HT,WT,AMTEMP,
                @IDUM7,AMPRES,IDUM8(2),TESTRT(3),
                @NREST,EXSTRT(3),NEXREC,RCSTRT(3),
                @NREC
0006             COMMON/X/MD(24)
          C      DATA MD/'JA','N ','FE','B ','MA','R ','AP','R ',
          C      @'MA','Y ','JU','N ','JU','L ','AU','G ','SE','P ',
          C      @'OC','T ','NO','V ','DE','C '/
          C       * * * PRINT PAGE HEADING * * *
          C      * * * USE TMX OR BX HEADER * * *
          C       ITSTYP(1)=BIKE
          C       ISTYP(2)=TMX
0007      170     IF(ITSTYP.EQ.1) GO TO 210
0009              TYPE 180
0010      180     FORMAT(48X,'EXERCISE RESPONSE TEST:  TREADMILL')
0011              GOTO 190
0012      210     TYPE 200
0013      200     FORMAT(48X,'EXERCISE RESPONSE TEST:  ERGOMETER')
0014      190     TYPE 220
0015      220     FORMAT(50X,'JSC CARDIOPULMONARY LABORATORY')
0016              TYPE 230
0017      230     FORMAT(56X,'30 SECOND SUMMARY')
0018              IMM=2*(ITDATE(1))
0019              IMM1=IMM-1
          C      * * * PRINT SUBJ INFO * * *
0020              TYPE 240,ICPIC,(NMSSEC(J),J=1,3),
                @ITDATE(2),MD(IMM1),MD(IMM),ITDATE(3),(UNIQ(I),I=1,2)
0021      240     FORMAT(/,' CPIC NO: ',I5,12X,
                @'SS NO: ',I3,'-',I2,'-',I4,8X,
                @'DATE OF TEST: ',I2,1X,2A2,I2,
                @10X,'UNIQUE NO: ',A2,I4)
0022              TYPE 250,TESTRT(1),TESTRT(2),TESTRT(3),WT,AGE
0023      250     FORMAT(1X,'START TIME OF TEST: ',I2,':',I2,':',I2,
                %14X,'WEIGHT(KG) : ',F5.1,
                %14X,'AGE (YRS) : ',I2)
          C      * * * PRINT COLUMN HEADINGS * * *
          C       IF TMX TEST, PRINT TMX HEADINGS
0024              IF(ITSTYP.EQ.1) GOTO 260
0026              TYPE 270
0027      270     FORMAT(//,'XMIN',5X,'HR',4X,'SBP',5X,'DBP',4X,'ELEV',
                @3X,'SPEED',4X,'\--- VO2 ----\',3X,'VCO2',5X,'M VOL',
                @3X,'RER',4X,'RESP',5X,'MV/VO2',2X,'MV/VCO2',3X,
                @'VO2/HR')
```

```
0028            TYPE 280
0029      280   FORMAT(' AVG',4X,'BPM',3X,'MM HG',3X,'MM HG',3X,
               @'% GRD',3X,'MPH',5X,'L/MN  ML/KG/MN',3X,'L/MIN',
               @4X,'L/MIN',10X,'RATE',22X,'ML/BEAT')
0030            GOTO 290
0031      260   TYPE 300
0032      300   FORMAT(//,'XMIN',5X,'HR',4X,'SBP',5X,'DBP',4X,'POWER',
               @2X,'WL SUM',3X,'\--- VO2 ----\',3X,'VCO2',5X,'M VOL',
               @3X,'RER',4X,'RESP',5X,'MV/VC2',2X,'MV/VCO2',3X,
               @'VO2/HR')
0033            TYPE 310
0034      310   FORMAT(' AVG',4X,'BPM',3X,'MM HG',3X,'MM HG',3X,
               @'WATTS',3X,'W-MIN',3X,'L/MN  ML/KG/MN',3X,'L/MIN',
               @4X,'L/MIN',10X,'RATE',22X,'ML/BEAT')
0035      290   CONTINUE
0036            RETURN
0037            END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

IMM      001400   INTEGER*2 VARIABLE
IMM1     001402   INTEGER*2 VARIABLE
J        001404   INTEGER*2 VARIABLE
I        001406   INTEGER*2 VARIABLE


COMMON BLOCK /PDATA/     LENGTH 000220

IDUM1    000000   INTEGER*2 VARIABLE
ISEX     000002   INTEGER*2 VARIABLE
NMSSEC   000004   INTEGER*2 ARRAY (3)
WLSUM    000012   INTEGER*2 VARIABLE
UNIQ     000014   INTEGER*2 ARRAY (2)
NMRTST   000020   INTEGER*2 VARIABLE
ICPID    000022   INTEGER*2 VARIABLE
IMODE    000024   INTEGER*2 VARIABLE
ITSTYP   000026   INTEGER*2 VARIABLE
IDUM3    000030   INTEGER*2 ARRAY (2)
IHR4     000034   INTEGER*2 VARIABLE
IHR8     000036   INTEGER*2 VARIABLE
IHR12    000040   INTEGER*2 VARIABLE
IHR16    000042   INTEGER*2 VARIABLE
IDUM4    000044   INTEGER*2 VARIABLE
IHRMAX   000046   INTEGER*2 VARIABLE
IDUM5    000050   INTEGER*2 VARIABLE
IBTHDY   000052   INTEGER*2 ARRAY (3)
AGE      000060   INTEGER*2 VARIABLE
NAME     000062   INTEGER*2 ARRAY (14)
ITDATE   000116   INTEGER*2 ARRAY (3)
IDUM6    000124   INTEGER*2 ARRAY (7)
HT       000142   REAL*4    VARIABLE
WT       000146   REAL*4    VARIABLE
AMTEMP   000152   REAL*4    VARIABLE
IDUM7    000156   INTEGER*2 VARIABLE
AMPRES   000160   REAL*4    VARIABLE
IDUM8    000164   INTEGER*2 ARRAY (2)
TESTRT   000170   INTEGER*2 ARRAY (3)
NREST    000176   INTEGER*2 VARIABLE
EXSTRT   000200   INTEGER*2 ARRAY (3)
NEXREC   000206   INTEGER*2 VARIABLE
RCSTRT   000210   INTEGER*2 ARRAY (3)
NREC     000216   INTEGER*2 VARIABLE

COMMON BLOCK /X/         LENGTH 000060

MD       000000   INTEGER*2 ARRAY (24)
```

```
FORTRAN IV       V01C-03A   FRI 29-FEB-80 14:47:14                  PAGE 001

0001              SUBROUTINE TIMSET
           C      PURPOSE: SET TIMERS TO KEEP TRACK OF TIME
           C      OF DAY WITHOUT CHANGING SYSTEM CLOCK
           C      WRITTEN BY WILLIAM CROSIER
           C      DATE 21-NOV-79
           C
0002              INTEGER HR,MN,SC
0003              REAL T0,T1
0004              COMMON/TIM/T1
0005              DATA T1/0./
0006       10     TYPE 110
0007       110    FORMAT(' TIME FROM THE TIME CODE GENERATOR/READER
                 %(HR:MN:SC) ?',$)
0008              ACCEPT 120,HR,MN,SC
0009       120    FORMAT(I2,1X,I2,1X,I2)
           C      GET NO. OF SECONDS SINCE SYSTEM BOOT OR CLOCK SET
0010              T0=SECNDS(0.)
0011              IF(HR.LT.0.OR.HR.GT.23.OR.MN.LT.0.OR.MN.GT.59.OR.
                 %SC.LT.0.OR.SC.GT.59) GO TO 10
0013              T1=T0-(FLOAT(SC)+60.*FLOAT(MN)+3600.*FLOAT(HR))
0014              RETURN
0015              END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

HR      000130  INTEGER*2 VARIABLE
MN      000132  INTEGER*2 VARIABLE
SC      000134  INTEGER*2 VARIABLE
T0      000136  REAL*4    VARIABLE
SECNDS  000000  REAL*4    PROCEDURE
FLOAT   000000  REAL*4    PROCEDURE

COMMON BLOCK /TIM/      LENGTH 000004

T1      000000  REAL*4    VARIABLE
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:47:23                 PAGE 001

        C
0001          SUBROUTINE TIMRC(HR,MN,SC)
        C     PURPOSE: READ TIME CF DAY, LSING SYSTEM CLOCK AND
        C     PREVIOUSLY CALCLLATED TIME CIFFERENCE BETWEEN
        C     SYSTEM CLOCK TIME AND TIME CODE GEN. TIME
        C
0002          INTEGER HR,MN,SC,REM
0003          COMMON/TIM/T1
0004          T=SECNCS(0.)-T1
0005          HR=IFIX(T/3600.)
0006          REM=IFIX(T-FLCAT(HR)*3600.)
0007          MN=REM/60
0008          SC=REM-(MN*60)
0009          RETURN
0010          END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

HR       000014  INTEGER*2 PARAMETER VARIABLE
MN       000016  INTEGER*2 PARAMETER VARIABLE
SC       000020  INTEGER*2 PARAMETER VARIABLE
REM      000026  INTEGER*2 VARIABLE
T        000030  REAL*4    VARIABLE
SECNDS   000000  REAL*4    PROCEDURE
IFIX     000000  INTEGER*2 PROCEDURE
FLOAT    000000  REAL*4    PROCEDURE

COMMON BLOCK /TIM/      LENGTH 000004

T1       000000  REAL*4    VARIABLE
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:50:23                PAGE 001

0001            SUBROUTINE LIGHT(L)
        C       SUBROUTINE LIGHT -- TURNS ON APPROPRIATE PANEL LIGHTS
0002            INTEGER DROUTB
0003            DROUTB = "167772
0004            L = IPEEK(DROUTB).AND."177400 .OR. L
0005            CALL IPOKE(DROUTB,L)
        D       TYPE 1,L
        D1      FORMAT(' L=',O7)
0006            RETURN
0007            END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

L       000014  INTEGER*2 PARAMETER VARIABLE
DROUTB  000016  INTEGER*2 VARIABLE
IPEEK   000000  INTEGER*2 PROCEDURE
IPOKE   000000  INTEGER*2 PROCEDURE
```

```
C   *****************************************************************
C   *****************************************************************
C   *                                                               *
C   *                   TECHNOLOGY INCORPORATED                     *
C   *                    LIFE SCIENCE DIVISION                      *
C   *                                                               *
C   *****************************************************************
C   *                                                               *
C   *   PROGRAM NAME:.................. REPORT                      *
C   *   AUTHOR:........................ ROY A. REED                 *
C   *   DATE:.......................... 2/NOV/78                    *
C   *                                                               *
C   *---------------------------------------------------------------*
C   *                                                               *
C   *   COMPUTER SYSTEM:.............. DEC PDP 1103 "LSI-11"        *
C   *   OPERATING SYSTEM:.............. RT-11  V01C-03A             *
C   *                                                               *
C   *---------------------------------------------------------------*
C   *                                                               *
C   *   COMPILING SEQUENCE:                                         *
C   *                                                               *
C   *      .R FORTRA <CR>                                           *
C   *      *REPORT=REPORT,DATA/U <CR>                               *
C   *      *<CR>                                                    *
C   *                                                               *
C   *---------------------------------------------------------------*
C   *                                                               *
C   *   RUN MODUAL LINKING SEQUENCE:                                *
C   *                                                               *
C   *      .R LINK <CR>                                             *
C   *      *REPORT=REPORT,DX0:SYSLIB/F <CR>                         *
C   *      *<CR>                                                    *
C   *                                                               *
C   *---------------------------------------------------------------*
C   *                                                               *
C   *   CALLING SEQUENCE:                                           *
C   *                                                               *
C   *      .R REPORT <CR>                                           *
C   *                                                               *
C   *---------------------------------------------------------------*
C   *                                                               *
C   *   PURPOSE:                                                    *
C   *                                                               *
C   *      PRINTS A HARD COPY OF THE MINUTE SUMMARY REPORT.         *
C   *      ALSO, PROVIDES THE USER AN EDITOR TO CHANGE INCORRECT    *
C   *      DATA.                                                    *
C   *                                                               *
C   *****************************************************************
C   *****************************************************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
              LOGICAL*1 YES,NO,IANS
              LOGICAL*4 OUT,IN
              INTEGER NMSSEC(3),NAME(14),ITDATE(3)
              INTEGER EXSTRT,IDATA(1536),RCSTRT
              INTEGER AGE,UNIQ(2),TESTRT,WLSUM
              REAL ADATE(3)
              REAL WT,RDATA(640)
              COMMON      COMMON /RDATA/IDATA
              EQUIVALENCE (IDATA(257),RDATA(1))
              EQUIVALENCE (IDATA(2),ISEX),(IDATA(3),NMSSEC(1))
              EQUIVALENCE (IDATA(6),WLSUM)
              EQUIVALENCE (IDATA(7),UNIQ(1)),(IDATA(9),NMRTST),(IDATA(10),ICPID)
              EQUIVALENCE (IDATA(11),IMODE),(IDATA(12),ITSTYP),(IDATA(25),AGE)
              EQUIVALENCE (IDATA(26),NAME(1))
              EQUIVALENCE (IDATA(40),ITDATE(1)),(IDATA(50),HT),(IDATA(52),WT)
              DATA YES/1HY/,NO/1HN/
              DATA OUT/'OUT '/,IN/'IN  '/
        C--
        C--  GET PATIENT FILE
        C--
        3     CALL DATA(IN)
        C--
        CC-- VERIFY CORRECT SUBJECT READ IN
        C--
              TYPE 4,(NMSSEC(I),I=1,3),UNIQ(1),UNIQ(2),NMRTST,ICPID
             %,(ITDATE(J),J=1,3),ITSTYP
        4     FORMAT(/,1X,'VERIFY CORRECT SUBJECT DATA READ FROM DISK',
             %//,1X,'S.S. NUM. ',I3,'-',I2,'-',I4,5X,'UNIQUE NO. ',A2,I4,
             %5X,'RETEST NO. ',I5,5X,'CPID NO. ',I4,5X,'TEST DATE ',
             %I2,'/',I2,'/',I2,5X,' TEST TYPE ',I2,
             %//,'$ IS THIS THE CORRECT SUBJECT FILE  ?? YES OR NO ?? ')
              ACCEPT 5,IANS
        5     FORMAT(A1)
              IF(IANS.NE.YES)GO TO 3
        C--
        C--  CHECT TO SEE IF EDIT IS REQUESTED
        C--
              TYPE 6
        6     FORMAT(/,'$  EDITOR REQUIRED  ?? YES OR NO ?? ')
              ACCEPT 5,IANS
              IF(IANS.EQ.YES) CALL EDIT
        C--
        C--  GET CURRENT DATE FROM SYSTEM
        C--
              CALL DATE(ADATE)
        C--
        C--  GET THE NUMBER OF REPORTS NEEDED
        C--
              TYPE 12
        12    FORMAT(/,'$ HOW MANY COPIES OF REPORT NEEDED ? ')
              ACCEPT 13,NUMRPT
        13    FORMAT(I2)
              IF(NUMRPT.LT.1)CALL EXIT
        C--
```

```
      C--  INSTRUCT THE USEF TC ALIGN THE PAGE
      C--
0037        TYPE 14
0038  14    FORMAT(/,1X,' PLEASE ALIGN TOP CF NEW PAGE WITH PRINTER HEAD',
           %/,' TYPE A "RETURN" WHEN READY ')
0039        ACCEPT 5,IANS
0040        DO 530 K=1,NUMRPT
0041        IWLS=0
0042        LINE=18
      C      * * * PRINT PAGE HEADING * * *
      C     * * * USE TMX OR BX HEADER * * *
0043        WLSUM=ICATA(6)
0044  170   IF(ITSTYP.EQ.1) GO TO 210
0046        TYPE 180
0047  180   FORMAT(48X,'EXEFCISE RESPCNSE TEST:  TREADMILL')
0048        GOTO 190
0049  210   TYPE 200
0050  200   FORMAT(48X,'EXEFCISE RESPCNSE_TEST:  ERGOMETER')
0051  190   TYPE 220
0052  220   FORMAT(50X,'JSC CARCIOPULMONARY LABCRATCRY')
0053        TYPE 230
0054  230   FORMAT(55X,'ONE MINUTE SUMMARY')
      C     * * * PRINT SUBJ INFC * * *
0055        TYPE 240,ICPIC,(NMSSEC(J),J=1,3),
           @(ITDATE(J),J=1,3),(UNIQ(I),I=1,2)
0056  240   FORMAT(/,' CPIC NO ',I5,12X,
           @'SS NO. ',I3,'-',I2,'-',I4,8X,
           @'DATE OF TEST ',I2,'/ ',I2,'/ ',
           @I2,10X,'UNIGUE NO ',A2,I4)
0057        TYPE 250,(IDATA(J),J=61,63),WT,AGE,WLSUM
0058  250   FORMAT(1X,'START TIME OF TEST: ',I2,':',I2,':',I2,
           %14X,'WEIGHT      (KG) : ',F5.1,
           %14X,'AGE (YRS) : ',I2,7X,'WCRK LCAC SUM ',I5)
      C     * * * PRINT CCLUMN HEADINGS * * *
      C      IF TMX TEST, PRINT TMX HEACINGS
0059        IF(ITSTYP.EQ.1) GOTC 260
0061        TYPE 270
0062  270   FORMAT(//,'XMIN',5X,'HR',4X,'SBP',5X,'DBP',4X,'ELEV',
           @3X,'SPEED',4X,'\--- VO2 ----\',3X,'VCO2',5X,'M VCL',
           @3X,'RER',4X,'RESP',5X,'MV/VC2',2X,'MV/VCO2',3X,
           @'VO2/HR')
0063        TYPE 280
0064  280   FORMAT(' AVG',4X,'BPM',3X,'MM HG',3X,'MM HG',3X,
           @'% GRD',3X,'MPH',5X,'L/MN  ML/KG/MN',3X,'L/MIN',
           @4X,'L/MIN',10X,'RATE',22X,'ML/BEAT')
0065        GOTO 290
0066  260   TYPE 300
0067  300   FORMAT(//,'XMIN',5X,'HR',4X,'SBP',5X,'DBP',4X,'POWER',
           @2X,'WL SUM',3X,'\--- VO2 ----\',3X,'VCO2',5X,'M VCL',
           @3X,'RER',4X,'RESP',5X,'MV/VC2',2X,'MV/VCO2',3X,
           @'VO2/HR')
0068        TYPE 310
0069  310   FORMAT(' AVG',4X,'BPM',3X,'MM HG',3X,'MM HG',3X,
           @'WATTS',3X,'W-MIN',3X,'L/MN  ML/KG/MN',3X,'L/MIN',
             %4X,'L/MIN',10X,'RATE',22X,'ML/BEAT')
0070   290    CONTINUE
0071          IPT=64
0072          IS=257
0073          IE=IS+3
0074          ISR=4
0075          DO 520 J=1,3
0076          WLS=0.
0077          IF(J.EQ.1)TYPE 500,(IDATA(IT),IT=61,63)
0079   500    FORMAT(/,1X,'* * * REST * * *    REAL TIME: ',I2,':',I2,':',I2)
0080          IF(J.EQ.2)TYPE 501,(IDATA(IT),IT=65,67)
0082   501    FORMAT(/,1X,'* * * EXERCISE * * *    REAL TIME: ',
             %I2,':',I2,':',I2)
0083          IF(J.EQ.3) TYPE 502,(IDATA(IT),IT=69,71)
0085   502    FORMAT(/,1X,'* * * RECOVERY * * *     REAL TIME: ',
             %I2,':',I2,':',I2)
0086          IF(IDATA(IPT).EQ.0)GO TO 515
0088          DO 510 I=1,IDATA(IPT)
0089          IF(IDATA(12).EQ.2)WLS=0.
0091          WL=0.
0092          VO2MKM=0.
0093          RER=0.
0094          RVMO2=0.
0095          RVMCO2=0.
0096          VO2HRB=0.
0097          IHR=IDATA(IS+1)
0098          IF(IDATA(12).EQ.1) WL=FLOAT(IDATA(IS+4))
0100          IF(IDATA(12).EQ.2) WL=FLOAT(IDATA(IS+4))/10.
0102          IF(IDATA(12).EQ.1)WLS=WLS+FLOAT(IDATA(IS+4))
0104          IF(IDATA(12).EQ.2)WLS=RDATA(ISR+3)
0106          O2=RDATA(ISR)
0107          CO2=RDATA(ISR+1)
0108          VOL=RDATA(ISR+2)
0109          IF(O2.GT.0)VO2MKM=(O2*1000.)/WT
0111          IF(O2.GT.0)RER=CO2/O2
0113          IRR=IDATA(IS+5)
0114          IF(O2.GT.0)RVMO2=VOL/O2
0116          IF(CO2.GT.0)RVMCO2=VOL/CO2
0118          IF(IHR.GT.0)VO2HRB=(O2*1000.)/FLOAT(IHR)
0120          TYPE 505,(IDATA(IP),IP=IS,IE),WL,WLS,O2,VO2MKM,CO2,VOL,RER,IRR
             %,RVMO2,RVMCO2,VO2HRB
0121   505    FORMAT(1X,I2,3(1X,I7),F8.1,F7.1,F8.2,F8.1,2X,F8.2,F8.1,F7.2,I7,
             %1X,3F9.1)
0122          IS=IS+20
0123          IE=IS+3
0124          ISR=ISR+10
0125   510    CONTINUE
0126   515    LINE=LINE+IDATA(IPT)
0127          IPT=IPT+4
0128   520    CONTINUE
0129          TYPE 521,(ADATE(ID),ID=1,3)
0130   521    FORMAT(/,94X,'REPORT DATE ',2A4,A1)
0131          IF(LINE.GT.66)LINE=LINE-66
0133          LINE=66-LINE
```

```
            DO 526,IL=1,LINE
            TYPE 525
      525   FORMAT(1X)
      526   CONTINUE
      530   CONTINUE
            CALL EXIT
            END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

ADATE    000006  REAL*4     ARRAY (3)
YES      000022  LOGICAL*1  VARIABLE
NO       000023  LOGICAL*1  VARIABLE
IANS     002612  LOGICAL*1  VARIABLE
OUT      000024  LOGICAL*4  VARIABLE
IN       000030  LOGICAL*4  VARIABLE
EXSTRT   002614  INTEGER*2  VARIABLE
RCSTRT   002616  INTEGER*2  VARIABLE
TESTRT   002620  INTEGER*2  VARIABLE
DATA     000000  REAL*4     PROCEDURE
I        002622  INTEGER*2  VARIABLE
J        002624  INTEGER*2  VARIABLE
EDIT     000000  REAL*4     PROCEDURE
DATE     000000  REAL*4     PROCEDURE
NUMRPT   002626  INTEGER*2  VARIABLE
EXIT     000000  REAL*4     PROCEDURE
K        002630  INTEGER*2  VARIABLE
IWLS     002632  INTEGER*2  VARIABLE
LINE     002634  INTEGER*2  VARIABLE
IPT      002636  INTEGER*2  VARIABLE
IS       002640  INTEGER*2  VARIABLE
IE       002642  INTEGER*2  VARIABLE
ISR      002644  INTEGER*2  VARIABLE
WLS      002646  REAL*4     VARIABLE
IT       002652  INTEGER*2  VARIABLE
WL       002654  REAL*4     VARIABLE
VO2MKM   002660  REAL*4     VARIABLE
RER      002664  REAL*4     VARIABLE
RVMO2    002670  REAL*4     VARIABLE
RVMCO2   002674  REAL*4     VARIABLE
VO2HRB   002700  REAL*4     VARIABLE
IHR      002704  INTEGER*2  VARIABLE
FLOAT    000000  REAL*4     PROCEDURE
O2       002706  REAL*4     VARIABLE
CO2      002712  REAL*4     VARIABLE
VOL      002716  REAL*4     VARIABLE
IRR      002722  INTEGER*2  VARIABLE
IP       002724  INTEGER*2  VARIABLE
ID       002726  INTEGER*2  VARIABLE
IL       002730  INTEGER*2  VARIABLE

COMMON BLOCK //           LENGTH 000004

COMMON   000000  REAL*4     VARIABLE

COMMON BLOCK /PDATA/      LENGTH 006000

IDATA    000000  INTEGER*2  ARRAY (1536)
RDATA    001000  REAL*4     ARRAY (640)
ISEX     000002  INTEGER*2  VARIABLE
NMSSEC   000004  INTEGER*2  ARRAY (3)
WLSUM    000012  INTEGER*2  VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

UNIQ    000014  INTEGER*2 ARRAY (2)
NMRTST  000020  INTEGER*2 VARIABLE
ICPID   000022  INTEGER*2 VARIABLE
IMODE   000024  INTEGER*2 VARIABLE
ITSTYP  000026  INTEGER*2 VARIABLE
AGE     000060  INTEGER*2 VARIABLE
NAME    000062  INTEGER*2 ARRAY (14)
ITDATE  000116  INTEGER*2 ARRAY (3)
HT      000142  REAL*4    VARIABLE
WT      000146  REAL*4    VARIABLE
```

```
        CC--
        C--  EDITOR SUBROOUTINE
        C--
        C--  ALLOWS THE USER TO CHANGE ANY OF THE SUBJECTS DATA
        C--  STORED IN THE SUBJECT'S DATA FILE.
        C--
0001          SUBROUTINE EDIT
        C--
        C--  SET UP VARIABLES
        C--
0002          LOGICAL*1 YES,NO,IRES
0003          LOGICAL*4 IN,OUT
0004          INTEGER IDATA(1536),IMIN(40)
0005          REAL RDATA(640)
0006          REAL*8 SECT(3)
        C--
        C-- SET UP COMMON BLOCK
        C--
0007          COMMON /PDATA/IDATA
        C--
        C--  SET UP EQUVIVALENCE STATEMENTS
        CC--
0008          EQUIVALENCE (IDATA(257),RDATA(1))
        C--
        C--  SET UP DATA VALUES
        C--
0009          DATA IN/'IN  '/,OUT/'OUT '/
0010          DATA YES/1HY/,NO/1HN/
0011          DATA SECT/'REST    ','EXERCISE','RECOVERY'/
        CC--
        C--  GET THE SECTION TO BE EDITED
        C--
0012    10    TYPE 100
0013    100   FORMAT(/,1X,'SELECT SECTION TO BE EDITED ',
             %/,5X,'1 - REST'
             %/,5X,'2 - EXERCISE',
             %/,5X,'3 - RECOVERY',
             %/,'$  TYPE IN THE NUMBER OF THE SECTION TO BE EDITED ')
0014          ACCEPT 200,ISEC
0015    200   FORMAT(I3)
0016          IF(ISEC.EQ.0)GO TO 1000
0018          IF(ISEC.LT.1.OR.ISEC.GT.3)GO TO 10
0020          MREC=IDATA(64+(ISEC-1)*4)
0021          ICNT=1
0022          TYPE 300
0023    300   FORMAT(/,' TYPE IN THE NUMBER(S) OF THE MINUTE(S) TO BE EDITED ')
0024    305   TYPE 310
0025    310   FORMAT('$  MINUTE ')
0026          ACCEPT 200,IMIN(ICNT)
0027          IF(IMIN(ICNT).EQ.0)GO TO 390
0029          IF(IMIN(ICNT).LE.MREC)GO TO 350
0031          TYPE 325
0032    325   FORMAT(/,1X,'MINUTE REQUESTED OUT OF RANGE: REQUEST IGNORED')
0033          GO TO 305
```

```
0034  350     ICNT=ICNT+1
0035          IF(ICNT.LT.40)GO TO 305
0037  390     ICNT=ICNT-1
0038          DO 500 IED=1,ICNT
0039          IF(ISEC.EQ.1)IPT=IMIN(IED)
0041          IF(ISEC.EQ.2)IPT=IMIN(IED)+IDATA(64)
0043          IF(ISEC.EQ.3)IPT=IMIN(IED)+IDATA(64)+IDATA(58)
0045          IS=257+(IPT-1)*20
0046          ISR=4+(IPT-1)*10
0047          TYPE 395,SECT(ISEC),IMIN(IED)
0048  395     FORMAT(//,1X,'* * * ',A8,' MINUTE ',I2,' * * *',/)
0049          TYPE 400
0050  400     FORMAT('$  HEART RATE ')
0051          ACCEPT 405,IRES
0052  405     FORMAT(A1)
0053          IF(IRES.NE.YES)GO TO 410
0055          TYPE 406,IDATA(IS+1)
0056  406     FORMAT('$  OLD = ',I3,' NEW = ')
0057          ACCEPT 407,IDATA(IS+1)
0058  407     FORMAT(I5)
0059  410     TYPE 415
0060  415     FORMAT('$  SYSTOLIC BLOOD PRESSURE ')
0061          ACCEPT 405,IRES
0062          IF(IRES.NE.YES)GO TO 420
0064          TYPE 406,IDATA(IS+2)
0065          ACCEPT 407,IDATA(IS+2)
0066  420     TYPE 425
0067  425     FORMAT('$  DIASTOLIC BLOOD PRESSURE ')
0068          ACCEPT 405,IRES
0069          IF(IRES.NE.YES)GO TO 430
0071          TYPE 406,IDATA(IS+3)
0072          ACCEPT 407,IDATA(IS+3)
0073  430     IF(IDATA(12).EQ.1)TYPE 435
0075  435     FORMAT('$  WORK LOAD ')
0076          IF(IDATA(12).EQ.2)TYPE 416
0078  416     FORMAT('$  ELEVATION ')
0079          ACCEPT 405,IRES
0080          IF(IRES.NE.YES)GO TO 440
0082          ELEV=FLOAT(IDATA(IS+4))/10.
0083          TYPE 417,ELEV
0084  417     FORMAT('$  OLD = ',F8.2,'  NEW = ')
0085          ACCEPT 418,ELEV
0086  418     FORMAT(F8.0)
0087          IDATA(IS+4)=IFIX(ELEV*10.)
0088  440     TYPE 445
0089  445     FORMAT('$  RESPIRATORY RATE ')
0090          ACCEPT 405,IRES
0091          IF(IRES.NE.YES)GO TO 450
0093          TYPE 406,IDATA(IS+5)
0094          ACCEPT 407,IDATA(IS+5)
0095  450     TYPE 455
0096  455     FORMAT(      '$  OXYGEN CONSUMPTION ')
0097          ACCEPT 405,IRES
0098          IF(IRES.NE.YES)GO TO 460
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:52:41                  PAGE 003

0100            TYPE 417,RDATA(ISR)
0101            ACCEPT 418,RDATA(ISR)
0102      460   TYPE 465
0103      465   FORMAT('$  CARBON DIOXIDE PRODUCTION ')
0104            ACCEPT 405,IRES
0105            IF(IRES.NE.YES)GO TO 470
0107            TYPE 417,RDATA(ISR+1)
0108            ACCEPT 418,RDATA(ISR+1)
0109      470   TYPE 475
0110      475   FORMAT('$  MINUTE VOLUMN ')
0111            ACCEPT 405,IRES
0112            IF(IRES.NE.YES)GO TO 480
0114            TYPE 417,RDATA(ISR+2)
0115            ACCEPT 418,RDATA(ISR+2)
0116      480   TYPE 485
0117      485   FORMAT('$  SPEED  MPH/RPM ')
0118            ACCEPT 405,IRES
0119            IF(IRES.NE.YES)GO TO 500
0121            TYPE 417,RDATA(ISR+3)
0122            ACCEPT 418,RDATA(ISR+3)
0123      500   CONTINUE
0124            GO TO 10
0125     1000   CALL DATA(OUT)
0126            RETURN
0127            END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IMIN    000014  INTEGER*2 ARRAY (40)
SECT    000134  REAL*8    ARRAY (3)
YES     000174  LOGICAL*1 VARIABLE
NO      000175  LOGICAL*1 VARIABLE
IRES    001310  LOGICAL*1 VARIABLE
IN      000164  LOGICAL*4 VARIABLE
OUT     000170  LOGICAL*4 VARIABLE
ISEC    001312  INTEGER*2 VARIABLE
MREC    001314  INTEGER*2 VARIABLE
ICNT    001316  INTEGER*2 VARIABLE
IED     001320  INTEGER*2 VARIABLE
IPT     001322  INTEGER*2 VARIABLE
IS      001324  INTEGER*2 VARIABLE
ISR     001326  INTEGER*2 VARIABLE
ELEV    001330  REAL*4    VARIABLE
FLOAT   000000  REAL*4    PROCEDURE
IFIX    000000  INTEGER*2 PROCEDURE
DATA    000000  REAL*4    PROCEDURE


COMMON BLOCK /PDATA/    LENGTH 006000

IDATA   000000  INTEGER*2 ARRAY (1536)
RDATA   001000  REAL*4    ARRAY (640)
```

```
C    *************************************************************
C    *************************************************************
C    *                                                           *
C    *                   TECHNOLOGY INCORPORATED                 *
C    *                   LIFE SCIENCE DIVISION                   *
C    *                                                           *
C    *************************************************************
C    *                                                           *
C    *   PROGRAM NAME:.................. PLOT                    *
C    *   AUTHOR:........................ CHARLES MANN            *
C    *          ........................ ROY A. REED             *
C    *   DATE:.......................... 2/NOV/78                *
C    *                                                           *
C    *-----------------------------------------------------------*
C    *                                                           *
C    *   COMPUTER SYSTEM:.............. DEC PDP 1103 "LSI-11"    *
C    *   OPERATING SYSTEM:............. RT-11  V01C-03A          *
C    *                                                           *
C    *-----------------------------------------------------------*
C    *                                                           *
C    *   COMPILING SEQUENCE:                                     *
C    *                                                           *
C    *      .R FORTRA <CR>                                       *
C    *      *PLOT=PLOT,DATA/U <CR>                               *
C    *      *SORT=SORT <CR>                                      *
C    *      *TITLES=TITLES <CR>                                  *
C    *      *GRAPH=GRAPH <CR>                                    *
C    *      *ONE=ONE <CR>                                        *
C    *      *TWO=TWO <CR>                                        *
C    *      *<CR>                                                *
C    *                                                           *
C    *-----------------------------------------------------------*
C    *                                                           *
C    *   RUN MODUAL LINKING SEQUENCE:                            *
C    *                                                           *
C    *      .R LINK <CR>                                         *
C    *      *PLOT=PLOT,SORT,TITLES,GRAPH,DX0:SYSLIB/F/C <CR>     *
C    *      *ONE,TWO                                             *
C    *      *<CR>                                                *
C    *                                                           *
C    *-----------------------------------------------------------*
C    *                                                           *
C    *   CALLING SEQUENCE:                                       *
C    *                                                           *
C    *      .R PLOT <CR>                                         *
C    *                                                           *
C    *-----------------------------------------------------------*
C    *                                                           *
C    *   PURPOSE:                                                *
C    *                                                           *
C    *      PRINTS A HARD COPY OF THE USER SELECTED GRAPHS       *
C    *                                                           *
C    *************************************************************
C    *************************************************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
C--------------------------------------------------------------
C       PLOT
C--------------------------------------------------------------
C
C       AUTHOR:  CHARLES MANN
C       DATE:    OCTOBER 19,1978
C       PURPOSE: TO DRIVE THE PLOTTING ROUTINES FOR CDAS
C
C--------------------------------------------------------------
C
        INTEGER IDATA(1536),CPID,ISS(3),IUNG(2),ITDATE(3)
        INTEGER IHR(49),SBP(49),DBP(49),ICOPY
        INTEGER WL(49)
        REAL IDATE(3)
        REAL VO2(49),VCO2(49),RDATA(640)
        INTEGER IYHR(49),IYVOL(49),IYWL(49),IYSBP(49)
        INTEGER IYDBP(49),IYVO2(49),IYVCO2(49),IYRESP(49)
        INTEGER IY1(49),IY2(49),IYO2WT(49)
        INTEGER*4 IN
        COMMON /PDATA/IDATA
        EQUIVALENCE(IDATA(257),RDATA(1))
        EQUIVALENCE(IDATA(10),CPID)
        EQUIVALENCE(IDATA(3),ISS(1))
        EQUIVALENCE(IDATA(7),IUNG(1))
        EQUIVALENCE(IDATA(12),IMODE)
        EQUIVALENCE(IDATA(40),ITDATE(1))
        EQUIVALENCE(IDATA(52),WT)
        EQUIVALENCE(IDATA(64),IRST)
        EQUIVALENCE(IDATA(68),IEXR)
        EQUIVALENCE(IDATA(72),IREC)
        DATA IN/'IN  '/
C  ZERO OUT PLOTTING BUFFER
        DO 10 I=1,49
         IHR(I)=0
         SBP(I)=0
         DBP(I)=0
         WL(I)=0
         VO2(I)=0.
         VCO2(I)=0.
10      CONTINUE
C  GET CURRENT DATE FROM THE SYSTEM
        CALL DATE(IDATE)
C  READ SUBJECT DATA IN FROM THE FILE
        CALL DATA(IN)
        ITOT=IRST+IEXR+IREC
        IF (ITOT.GT.49) ITOT=49
        DO 20 I=1,ITOT
         IHR(I)=IDATA(258+(I-1)*20)
         SBP(I)=IDATA(259+(I-1)*20)
         DBP(I)=IDATA(260+(I-1)*20)
         WL(I)=IDATA(261+(I-1)*20)
         VO2(I)=RDATA(4+(I-1)*10)
         VCO2(I)=RDATA(5+(I-1)*10)
20      CONTINUE
```

```
C       TYPE 800,WL,ICUM,VC2,RDUM,VCO2,RDUM
0800    FORMAT(5(10(1X,I7),/),//,5(10(1X,F7.2),/),
C    *     //,5(10(1X,F7.2),/))
C  PLACE DATA INTO ARRAYS SCALED FROM 0 TO 50
        DO 100 I=1,49
          IYHR(I)=(IDATA(258+(I-1)*20)+2)/4
          IYVOL(I)=IFIX((RDATA(6+(I-1)*10)/4.)+.5)
          IYWL(I)=IFIX((FLOAT(IDATA(261+(I-1)*20))/6.)+.5)
          IYSBP(I)=(IDATA(259+(I-1)*20)+3)/6
          IYDBP(I)=(IDATA(260+(I-1)*20)+3)/6
          IYVO2(I)=IFIX((RDATA(4+(I-1)*10))*10.)
          IYO2WT(I)=IFIX(((RDATA(4+(I-1)*10)*1000.)/WT)+.5)
          IYVCO2(I)=IFIX((RDATA(5+(I-1)*10))*10.)
          IYRESP(I)=(IDATA(262+(I-1)*20)+1)/2
100     CONTINUE
C  PRINT THE CHOICES
110     TYPE 105
105     FORMAT('0 0--NO PLOTS (STOP)',T30,' 7--SBP VS WL',/,
     *     '  1--REGULAR 4 PLOTS',T30,' 8--SBP VS HR',/,
     *     '  2--MIN. VOL. VS TIME',T30,' 9--SBP VS VO2',/,
     *     '  3--RESP. RATE VS TIME',T30,'10--VCO2 VS VO2',/,
     *     '  4--VO2 VS TIME(ML/KG-MIN)',T30,'11--MIN. VOL. VS VO2',/,
     *     '  5--VO2 VS WL',T30,'12--MIN. VOL. VS VCO2',/,
     *     '  6--HR VS WL',//,' MAKE A SELECTION--',$)
C  READ THE 1ST CHOICE
        ACCEPT 115,IPICK1
115     FORMAT(I2)
        IF (IPICK1.LE.0) STOP
        IF (IPICK1.GT.12) GO TO 110
        IF (IPICK1.EQ.1) GO TO 300
C  ASK FOR SECOND CHOICE
        TYPE 125
125     FORMAT('OMAKE A SECOND SELECTION--',$)
        ACCEPT 115, IPICK2
C  ASK FOR THE NUMBER OF COPIES WANTED
        TYPE 305
        ACCEPT 315,ICOPY
        TYPE 25
        ACCEPT 26,I
C--THIS SECTION MOVES THE CORRECT DATA INTO THE FIRST PLOT ARRAY, IY1
        GO TO (110,200,210,220,230,240,250,260,270,280,290,298),IPICK1
200     CALL SORT(0,IYVOL,IYVOL,IY1)
        GO TO 400
210     CALL SORT(0,IYRESP,IYRESP,IY1)
        GO TO 400
220     CALL SORT(0,IYO2WT,IYO2WT,IY1)
        GO TO 400
230     CALL SORT(1,IYWL,IYVO2,IY1)
        GO TO 400
240     CALL SORT(1,IYWL,IYHR,IY1)
        GO TO 400
250     CALL SORT(1,IYWL,IYSBP,IY1)
        GO TO 400
260     CALL SORT(1,IYHR,IYSBP,IY1)
```

```
              GO TO 400
        270   CALL SORT(1,IYVC2,IYSBP,IY1)
              GO TO 400
        280   CALL SORT(1,IYVC2,IYVCO2,IY1)
              GO TO 400
        290   CALL SORT(1,IYVC2,IYVOL,IY1)
              GO TO 400
        298   CALL SORT(1,IYVCO2,IYVOL,IY1)
        400   GO TO (110,406,410,420,430,440,450,460,470,480,490,498),IPICK2
        406   CALL SORT(0,IYVCL,IYVOL,IY1)
              GO TO 500
        410   CALL SORT(0,IYRESP,IYRESP,IY2)
              GO TO 500
        420   CALL SORT(0,IYO2WT,IYO2WT,IY2)
              GO TO 500
        430   CALL SORT(1,IYWL,IYVO2,IY2)
              GO TO 500
        440   CALL SORT(1,IYWL,IYHR,IY2)
              GO TO 500
        450   CALL SORT(1,IYWL,IYSBP,IY2)
              GO TO 500
        460   CALL SORT(1,IYHR,IYSBP,IY2)
              GO TO 500
        470   CALL SORT(1,IYVC2,IYSBP,IY2)
              GO TO 500
        480   CALL SORT(1,IYVC2,IYVCO2,IY2)
              GO TO 500
        490   CALL SORT(1,IYVC2,IYVOL,IY2)
              GO TO 500
        498   CALL SORT(1,IYVCO2,IYVOL,IY2)
        500   CONTINUE
        510   CALL GRAPH(IPICK1,IPICK2,IY1,IY2,ICPID,ISS,
             *    ILNG,ITDATE,IDATE)
              TYPE 35
              ICOPY=ICOPY-1
              IF (ICOPY.GT.0) GO TO 510
              GO TO 110
        C  FIND OUT HOW MANY COPIES ARE WANTED
        300   TYPE 305
        305   FORMAT('0HOW MANY COPIES DO YOU WANT?--',$)
              ACCEPT 315,ICOPY
        315   FORMAT(I2)
              TYPE 25
        25    FORMAT(' ALIGN NEW PAGE WITH PRINT HEAD.',
             *    '  HIT RETURN KEY TO PROCEED.')
              ACCEPT 26,I
        26    FORMAT(I1)
        30    CONTINUE
              CALL ONE(CPID,ISS,ILNG,ITDATE,IHR,SBP,
             *    DBP,IDATE,ICUM)
              TYPE 35
        35    FORMAT('0')
              CALL TWO(CPID,ISS,ILNG,ITDATE,WL,VC2,
             *    VCO2,IDATE,ICUM,IMODE,WT)
```

```
0135            ICOPY=ICOPY-1
0136            TYPE 35
0137            IF (ICOPY.GT.0) GO TO 30
0139            GO TO 110
0140            END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

IHR      000006  INTEGER*2 ARRAY (49)
SBP      000150  INTEGER*2 ARRAY (49)
DBP      000312  INTEGER*2 ARRAY (49)
WL       000454  INTEGER*2 ARRAY (49)
IDATE    000616  REAL*4    ARRAY (3)
VO2      000632  REAL*4    ARRAY (49)
VCO2     001136  REAL*4    ARRAY (49)
IYHR     001442  INTEGER*2 ARRAY (49)
IYVOL    001604  INTEGER*2 ARRAY (49)
IYWL     001746  INTEGER*2 ARRAY (49)
IYSBP    002110  INTEGER*2 ARRAY (49)
IYDBP    002252  INTEGER*2 ARRAY (49)
IYVO2    002414  INTEGER*2 ARRAY (49)
IYVCO2   002556  INTEGER*2 ARRAY (49)
IYRESP   002720  INTEGER*2 ARRAY (49)
IY1      003062  INTEGER*2 ARRAY (49)
IY2      003224  INTEGER*2 ARRAY (49)
IYO2WT   003366  INTEGER*2 ARRAY (49)
ICOPY    004442  INTEGER*2 VARIABLE
IN       003530  INTEGER*4 VARIABLE
I        004444  INTEGER*2 VARIABLE
DATE     000000  REAL*4    PROCEDURE
DATA     000000  REAL*4    PROCEDURE
ITOT     004446  INTEGER*2 VARIABLE
IFIX     000000  INTEGER*2 PROCEDURE
FLOAT    000000  REAL*4    PROCEDURE
IPICK1   004450  INTEGER*2 VARIABLE
IPICK2   004452  INTEGER*2 VARIABLE
SORT     000000  REAL*4    PROCEDURE
GRAPH    000000  REAL*4    PROCEDURE
ICPID    004454  INTEGER*2 VARIABLE
ONE      000000  REAL*4    PROCEDURE
IDUM     004456  INTEGER*2 VARIABLE
TWO      000000  REAL*4    PROCEDURE


COMMON BLOCK /PDATA/    LENGTH 006000

IDATA    000000  INTEGER*2 ARRAY (1536)
RDATA    001000  REAL*4    ARRAY (640)
CPID     000022  INTEGER*2 VARIABLE
ISS      000004  INTEGER*2 ARRAY (3)
IUNG     000014  INTEGER*2 ARRAY (2)
IMODE    000026  INTEGER*2 VARIABLE
ITDATE   000116  INTEGER*2 ARRAY (3)
WT       000146  REAL*4    VARIABLE
IRST     000176  INTEGER*2 VARIABLE
IEXR     000206  INTEGER*2 VARIABLE
IREC     000216  INTEGER*2 VARIABLE
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:56:20                  PAGE 001

0001          SUBROUTINE DATA(IOSTAT)
0002          LOGICAL*1 IFILE(12)
0003          INTEGER FILE(4)
      C
0004          INTEGER*4 IOSTAT,OUT
      C
      C
0005          COMMON /PDATA/ IPATNT(1536) ! THIS IS PATIENT DATA.
      C
      C
0006          DATA OUT/'OUT '/
0007          DATA IFILE/'D','X','1',' ',' ',' ',' ',' ',' ','D','A','T'/
      C
0008          IF(IQSET(2).NE.0) STOP 'QUEUE ALLOCATION FALIURE'
0010          IF(IOSTAT.EQ.OUT)GO TO 10
0012          TYPE 1025
0013    1025  FORMAT('$  PLEASE TYPE PATIENT DATA FILE NAME (UP TO 6 SYMBOLS):')
0014          ACCEPT 1030,(IFILE(I),I=4,9)
0015    1030  FORMAT(6A1)
      C
      C
      C
0016    10    CONTINUE
      C
      C--CONVERT ASCII FILE NAME TO RADIX 50
      C
0017          IF(IOSTAT.NE.OUT) CALL IRAD50(12,IFILE,FILE)
      C
      C--OPEN FILE:
      C
0019          ICHAN =IGETC()
0020          IF (ICHAN .LT. 0) STOP 'NO AVAILABLE CHANNEL'
0022          IF (LOOKUP(ICHAN,FILE) .LT. 0) STOP 'FILE ALLOCATION FAILURE'
      C
      C--NOW READS OR WRITES  DATA:
      C
0024          IF(IOSTAT.EQ.OUT) GO TO 1050
0026          IF(IREADW(1536,IPATNT,0,ICHAN).LT.0) STOP 'DISK READ ERROR'
0028          GO TO 1060
0029    1050  CONTINUE
0030          IF(IWRITW(1536,IPATNT,0,ICHAN).LT.0) STOP 'DISK WRITE ERROR'
0032    1060  CONTINUE
0033          CALL CLOSEC(ICHAN)
0034          RETURN
0035          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IFILE   000016  LOGICAL*1 ARRAY (12)
FILE    000032  INTEGER*2 ARRAY (4)
IOSTAT  000014  INTEGER*4 PARAMETER VARIABLE
OUT     000042  INTEGER*4 VARIABLE
IQSET   000000  INTEGER*2 PROCEDURE
I       000330  INTEGER*2 VARIABLE
IRAD50  000000  INTEGER*2 PROCEDURE
ICHAN   000332  INTEGER*2 VARIABLE
IGETC   000000  INTEGER*2 PROCEDURE
LOOKUP  000000  INTEGER*2 PROCEDURE
IREADW  000000  INTEGER*2 PROCEDURE
IWRITW  000000  INTEGER*2 PROCEDURE
CLOSEC  000000  REAL*4    PROCEDURE

COMMON BLOCK /PDATA/    LENGTH 006000

IPATNT  000000  INTEGER*2 ARRAY (1536)
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:56:41                PAGE 001

      C*********************************************************
      C
0001          SUBROUTINE ONE(CPID,SS,UNQ,ITDATE,IHR,ISBP,
             @                IDBP,IDATE,ICOPY)
      C
      C       THIS SUBPROGRAM TAKES ARGUMENTS PASSED
      C       FROM MAIN PROGRAM "PLOT" ANDGENERATES
      C       GRAPHS OF HEART RATE VS TIME AND BLOOD
      C       PRESSURES VS TIME ON ONE PAGE OF 15
      C       INCH PRINTOUT PAPER.
      C
      C       TIME IS A CONSTANT 50 MINUTES.
      C       ARGUMENTS PASSED:
      C       CPID                CARDIOPULMONARY ID NO.
      C       SS             SOCIAL SECURITY NUMBER
      C       UNQ            UNIQUE TEST NUMBER
      C       ITDATE                DATE OF TEST
      C       IHR            HEART RATE (BPM)
      C       SBP            SYSTOLIC BLOOD PRES.
      C       DBP            DIASTOLIC BLOOD PRES.
      C       ICOPY                 COPIES OF GRAPH REQ'D
      C       IDATE                 REPORT(TODAY'S) DATE
      C
      C
      C       RUTT   JUN 78
      C
      C*********************************************
0002          DIMENSION IDATE(3),IYLABL(11),IBAR(11),
             @IHR(49),I1DASH(49),I1BLNK(49),I2DASH(49),
             @I2BLNK(49),ITIME(50),ISBP(49),IDBP(49),
             @ILIN(49)
0003          REAL IDATE
0004          INTEGER IYLBL2(11),SS(3),ITDATE(3),UNQ(2),CPID
0005          DATA ITIME/18*1H ,1HT,1H ,1HI,1H ,
             @1HM,1H ,1HE,25*1H /
0006          DATA ICHAR1/1H-/
0007          DATA ICHAR2/1H /
0008          DATA IBAR/11*1H\/
0009          DATA I1BLNK/49*' '/
0010          DATA I2BLNK/49*' '/
0011          DATA I1DASH/49*'-'/
0012          DATAI2DASH/49*'-'/
0013          DATA IO/1HO/
0014          DATA IX/1HX/
0015          DATA ISTAR/1H*/
0016          DATA IYLABL/0,30,60,90,120,150,180,
             @210,240,270,300/
0017          DATA IYLBL2/0,20,40,60,80,100,
             @    120,140,160,180,200/
0018          DATA ILIN/'  ','  ','  ','  ',' 5',
             @'  ','  ','  ','  ','10',
             @'  ','  ','  ','  ','15',
             @'  ','  ','  ','  ','20',
             @'  ','  ','  ','  ','25',
                @'  ','  ','  ','  ','30',
                @'  ','  ','  ','  ','35',
                @'  ','  ','  ','  ','40',
                @'  ','  ','  ','  ','45',
                @'  ','  ','  ','  '/
 0019    10      ICOPY=ICOPY-1
         C        * * * PRINT PAGE HEADING * * *
 0020            TYPE 951
 0021    951     FORMAT(15X,'JSC CARDIOPULMONARY LABORATORY',
                @45X,'JSC CARDIOPULMONARY LABORATORY')
         C      * * * PRINT GRAPH TITLES * * *
 0022            TYPE 900
 0023    900     FORMAT(/,18X,'SBP(X) & DBP(O) VS TIME',51X,
                @'HEART RATE VS TIME',/)
 0024            TYPE 910
 0025    910     FORMAT(18X,'BLOOD PRESSURE (MM HG)',52X,
                @'HEART RATE (BPM)')
         C      * * * PRINT Y LABELS * * *
 0026            TYPE 920,IYLABL,IYLBL2
 0027    920     FORMAT(6X,I1,4X,I2,3X,I2,3X,I2,
                @2X,I3,2X,I3,2X,I3,2X,I3,2X,I3,2X,I3,
                @2X,I3,19X,I2,4X,I2,3X,I2,
                @3X,I2,2X,I3,2X,I3,2X,I3,2X,I3,
                @2X,I3,2X,I3,2X,I3)
         C      * * * PRINT Y LABEL MARKERS * * *
 0028            TYPE 927,IBAR,IBAR
 0029    927     FORMAT(2X,11(4X,1A1),17X,11(4X,1A1))
         C      * * * PRINT TOP LINE OF GRAPHS * * *
 0030            TYPE 928,IBAR(1),I1DASH,IBAR(1),
                @IBAR(1),I1DASH,IBAR(1)
 0031    928     FORMAT(5X,'0',A1,49A1,A1,
                @20X,'0',A1,49A1,A1)
         C      * * * SET UP THE PLOT * * *
 0032            DO 100 I=1,49
 0033            IMULT=0
         C      * * * CHECK IF LINE MULTIPLE OF 5 * * *
         C       DIVIDE BY FIVE AND TEST.
         C       EVEN MULTIPLE. THEN USE DASHES IN
         C       LINE ARRAY INSTEAD OF BLANKS.
         C       IF REMAINDER EQUAL 0, LINE IS
 0034            XLIN=FLOAT(I)/5.
 0035            IXLIN=XLIN
 0036            XLDIF=XLIN-IXLIN
 0037            IF(XLDIF.GT.0.01) GOTO 140
         C      * * * SET LINE ARRAYS=DASHES * * *
 0039            DO 200 JJ=1,49
 0040            I1BLNK(JJ)=ICHAR1
 0041    200     I2BLNK(JJ)=ICHAR1
         C      * * * SCALE HR TO FIT 50 SPACES * * *
         C       AND ROUND OFF
         C       IF IHR=0 DO NOT PRINT DATA POINT
         C       PRINT 49 BLANKS OR 49 DASHES
         C       OTHERWISE INSERT AN X TO REPRESENT
         C       HR AVG FOR THAT MINUTE.
         C     * * * TEST IF IHR=0 * * *
0042     140    IF(IHR(I).LT.0) GOTO 150
0044            XDIS=(IHR(I)+2)/4
0045            IXDIS=XDIS
0046            XDIF=XDIS-IXDIS
0047            IF(XDIF.GT.0.5) IXDIS=IXDIS+1
0049            J=IXDIS
0050            I1BLNK(J)=IX
         C     * * * SCALE SBP TO FIT 49 SPACES * * *
         C      ROUNDOFF. IF SBP=0 DO NOT PRINT A DATA POINT.
         C      PRINT 49 BLANKS OR 49 DASHES.
         C      INSERT AN "X" IN THE LINE TO REPRESENT
         C      SBP FOR THAT MINUTE.
         C     * * * TEST FOR SBP=0 * * *?
0051     150    IF(ISBP(I).LT.0) GOTO 160
0053            XDIS1=(ISBP(I)+3)/6
0054            IXD1=XDIS1
0055            XDF1=XDIS1-IXD1
0056            IF(XDF1.GT.0.5) IXD1=IXD1+1
0058            J1=IXD1
0059            I2BLNK(J1)=IX
         C     * * * SCALE DBP TO FIT 49 SPACES * * *
         C      ROUNDOFF. IF DBP=0 DO NOT PRINT A DATA POINT.
         C      PRINT 49 DASHES OR 49 BARS.
         C      INSERT AN X IN THE LINE TO REPRESENT
         C      DBP FOR THAT MINUTE.
         C     * * * TEST FOR DBP=0 * * *
0060     160    IF(IDBP(I).LT.0) GOTO 180
0062            XDIS2=(IDBP(I)+3)/6
0063            IXD2=XDIS2
0064            XDF2=XDIS2-IXD2
0065            IF(XDF2.GT.0.5) IXD2=IXD2+1
0067            J2=IXD2
0068            I2BLNK(J2)=IO
         C     * * * TEST IF SBP=DBP * * *
0069            IF(J1.NE.J2) GOTO 180
0071            I2BLNK(J2)=ISTAR
         C     * * * LETS PRINT A DATA LINE * * *
0072     180    TYPE 190,ITIME(I),ILIN(I),IBAR(1),I2BLNK,
               @IBAR(1),ITIME(I),ILIN(I),IBAR(1),I1BLNK,IBAR(1)
0073     190    FORMAT(1X,A1,2X,A2,A1,49A1,A1,16X,A1,2X,
               @A2,A1,49A1,A1)
         C     * * * RESTORE LINE ARRAYS * * *
0074            DO 300 JJJ=1,49
0075            I1BLNK(JJJ)=ICHAR2
0076     300    I2BLNK(JJJ)=ICHAR2
0077     100    CONTINUE
         C     * * * NOW ADD THE BOTTOM LINE * * *
0078            TYPE 195,IBAR(1),I1DASH,IBAR(1),
               @IBAR(1),I1DASH,IBAR(1)
0079     195    FORMAT(4X,'50',A1,49A1,A1,19X,'50',
               @A1,49A1,A1)
         C     * * * OK . . .  TACK ON THE SUBJ INFO * * *
0080            TYPE 800,CPIC,CPIC,SS,SS,LNG,LNG,ITDATE,IDATE
```

```
                &    ,ITCATE,ICATE
0081  800        FORMAT(/,5X,'CPID       :',I7,53X,
                @'CPID       :',I7,/,5X,'SS NUMBER : ',
                @I3,'-',I2,'-',I4,48X,'SS NUMBER : ',I3,'-',I2,'-',I4,
                @/,5X,'UNIQUE NC.: ',A2,I4,53X,'UNIGUE
                @ NO.: ',A2,I4,/,5X,'TEST CATE : ',I2,'/',I2,'/',I2,
                @8X,'REPORT DATE : ',2A4,A1,
                @19X,'TEST DATE : ',I2,'/',I2,'/',I2,
                @8X,'REPORT DATE : ',2A4,A1)
0082             TYPE 810
0083  810        FORMAT(20X,'NOTE: AN ASTERISK (*)
                @PLOTTED ON ANY GRAPH INDICATES BOTH
                @ VARIABLES EGUAL AT THAT MINUTE')
0084             RETURN
0085             END
```

```
FORTRAN IV      STORAGE MAP

NAME      OFFSET  ATTRIBUTES

IDATE     000032  REAL*4    PARAMETER ARRAY (3)
IYLABL    000036  INTEGER*2 ARRAY (11)
IBAR      000064  INTEGER*2 ARRAY (11)
IHR       000024  INTEGER*2 PARAMETER ARRAY (49)
I1DASH    000112  INTEGER*2 ARRAY (49)
I1BLNK    000254  INTEGER*2 ARRAY (49)
I2DASH    000416  INTEGER*2 ARRAY (49)
I2BLNK    000560  INTEGER*2 ARRAY (49)
ITIME     000722  INTEGER*2 ARRAY (50)
ISBP      000026  INTEGER*2 PARAMETER ARRAY (49)
IDBP      000030  INTEGER*2 PARAMETER ARRAY (49)
ILIN      001066  INTEGER*2 ARRAY (49)
IYLBL2    001230  INTEGER*2 ARRAY (11)
SS        000016  INTEGER*2 PARAMETER ARRAY (3)
ITDATE    000022  INTEGER*2 PARAMETER ARRAY (3)
UNQ       000020  INTEGER*2 PARAMETER ARRAY (2)
CPID      000014  INTEGER*2 PARAMETER VARIABLE
ICOPY     000034  INTEGER*2 PARAMETER VARIABLE
ICHAR1    001256  INTEGER*2 VARIABLE
ICHAR2    001260  INTEGER*2 VARIABLE
IO        001262  INTEGER*2 VARIABLE
IX        001264  INTEGER*2 VARIABLE
ISTAR     001266  INTEGER*2 VARIABLE
I         002556  INTEGER*2 VARIABLE
IMULT     002560  INTEGER*2 VARIABLE
XLIN      002562  REAL*4    VARIABLE
FLOAT     000000  REAL*4    PROCEDURE
IXLIN     002566  INTEGER*2 VARIABLE
XLDIF     002570  REAL*4    VARIABLE
JJ        002574  INTEGER*2 VARIABLE
XDIS      002576  REAL*4    VARIABLE
IXDIS     002602  INTEGER*2 VARIABLE
XDIF      002604  REAL*4    VARIABLE
J         002610  INTEGER*2 VARIABLE
XDIS1     002612  REAL*4    VARIABLE
IXD1      002616  INTEGER*2 VARIABLE
XDF1      002620  REAL*4    VARIABLE
J1        002624  INTEGER*2 VARIABLE
XDIS2     002626  REAL*4    VARIABLE
IXD2      002632  INTEGER*2 VARIABLE
XDF2      002634  REAL*4    VARIABLE
J2        002640  INTEGER*2 VARIABLE
JJJ       002642  INTEGER*2 VARIABLE
```

```
      C*********************************************************
      C
0001        SUBROUTINE TWC(CPID,SS,UNQ,ITDATE,WL,VO2,
           @             VCO2,IDATE,ICOPY,IMODE,WT)
      C
      C     THIS SUBPROGRAM TAKES ARGUMENTS PASSED
      C     FROM MAIN PROGRAM "PLOT" AND GENERATES
      C     GRAPHS OF WORK LOAD VS TIME AND GASES
      C     VO2 & VCO2 VS TIME ON ONE PAGE OF 15
      C     INCH PRINTOUT PAPER.
      C
      C     TIME IS A CONSTANT 50 MINUTES.
      C     ARGUMENTS PASSED:
      C     CPID                    CARDIOPULMONARY ID NO.
      C     SS             SOCIAL SECURITY NUMBER
      C     UNQ            UNIQUE TEST NUMBER
      C     ITDATE                  DATE OF TEST
      C     WL             WORK LOAD (WATTS)
      C     VO2            OXYGEN CONSUMPTION (L/MIN)
      C     VCO2                    CARBON DIOXIDE PRODUCTION (L/MIN)
      C     ICOPY                   COPIES OF GRAPH REQ'D
      C     IDATE                   REPORT(TODAY'S) DATE
      C     IMODE                   BICYCLE OR TREADMILL
      C
      C
      C     RUTT   JUN 78
      C
      C*********************************************************
0002        DIMENSION IDATE(3),IYLABL(11),IBAR(11),
           @WL(49),I1DASH(49),I1BLNK(49),I2DASH(49),
           @I2BLNK(49),ITIME(50),VO2(49),VCO2(49),
           @ILIN(49),IFIV(22)
0003        INTEGER IYLABX(11)
0004        REAL IDATE
0005        INTEGER WL,SS(3),UNQ(2),ITDATE(3),CPID
0006        DATA ITIME/18*1H ,1HT,1H ,1HI,1H ,
           @1HM,1H ,1HE,25*1H /
0007        DATA ICHAR1/1H-/
0008        DATA ICHAR2/1H /
0009        DATA IBAR/11*1H\/
0010        DATA I1BLNK/49*' '/
0011        DATA I2BLNK/49*' '/
0012        DATA I1DASH/49*'-'/
0013        DATA I2DASH/49*'-'/
0014        DATA IO/1HO/
0015        DATA IX/1HX/
0016        DATA ISTAR/1H*/
0017        DATA IYLABL/0,30,60,90,120,150,180,
           @210,240,270,300/
0018        DATA  IYLABX/0,8,16,24,32,40,48,
           @56,64,72,80/
0019        DATA IFIV/'0.','0 ','0.','5 ',
           @'1.','0 ','1.','5 ','2.',
           @'0 ','2.','5 ','3.','0 ',
                     @'3.','5 ','4.','0 ',
                     @'4.','5 ','5.','0 '/
0020                  DATA ILIN/'  ','  ','  ','  ',' 5',
                     @'  ','  ','  ','  ','10',
                     @'  ','  ','  ','  ','15',
                     @'  ','  ','  ','  ','20',
                     @'  ','  ','  ','  ','25',
                     @'  ','  ','  ','  ','30',
                     @'  ','  ','  ','  ','35',
                     @'  ','  ','  ','  ','40',
                     @'  ','  ','  ','  ','45',
                     @'  ','  ','  ','  '/
0021      10          ICOPY=ICOPY-1
          C            * * * PRINT PAGE HEADING * * *
0022                  TYPE 951
0023      951         FORMAT(15X,'JSC CARDIOPULMONARY LABORATORY',
                     @45X,'JSC CARDIOPULMONARY LABORATORY')
          C           * * * PRINT GRAPH TITLES * * *
0024                  IF(IMODE.EQ.1)TYPE 900
0026      900         FORMAT(/,17X,'VC2(X) & VCC2(0) VS TIME',54X,
                     @'WORK LOAD VS TIME',/)
0027                  IF(IMODE.EQ.2)TYPE 901
0029      901         FORMAT(/,17X,'VC2(X) & VCC2(0) VS TIME',54X,
                     @'VO2 ML/KG VS TIME',/)
0030                  IF(IMODE.EQ.1)TYPE 910
0032      910         FORMAT(21X,'GASES (L / MIN)',59X,
                     @'WORK LOAD (WATTS)')
0033                  IF(IMODE.EQ.2)TYPE 911
0035      911         FORMAT(21X,'GASES (L / MIN)',59X,
                     @'VO2 ML. / KG. BODY WEIGHT')
          C           * * * PRINT Y LABELS * * *
0036                  IF(IMODE.EQ.1)TYPE 920,IFIV,IYLABL
0038      920         FORMAT(5X,2A2,10(1X,2A2),
                     @18X,I2,4X,I2,3X,I2,
                     @3X,I2,2X,I3,2X,I3,2X,I3,2X,I3,
                     @2X,I3,2X,I3,2X,I3)
0039                  IF(IMODE.EQ.2)TYPE 921,IFIV,IYLABX
0041      921         FORMAT(5X,2A2,10(1X,2A2),18X,
                     @I2,10(3X,I2))
          C           * * * PRINT Y LABEL MARKERS * * *
0042                  TYPE 927,IBAR,IBAR
0043      927         FORMAT(2X,11(4X,1A1),17X,11(4X,1A1))
          C           * * * PRINT TOP LINE OF GRAPHS * * *
0044                  I1DASH(49)=ICHAR1
0045                  TYPE 928,IBAR(1),I1DASH,IBAR(1),
                     @IBAR(1),I1DASH,IBAR(1)
0046      928         FORMAT(5X,'0',A1,49A1,A1,
                     @20X,'0',A1,49A1,A1)
          C           * * * SET UP THE PLOT * * *
0047                  DO 100 I=1,49
0048                  IMULT=0
          C           * * * CHECK IF LINE MULTIPLE OF 5 * * *
          C            DIVIDE BY FIVE AND TEST.
          C            EVEN MULTIPLE, THEN USE DASHES IN
```

```
        C      LINE ARRAY INSTEAD OF BLANKS.
        C      IF REMAINDER EQUAL 0, LINE IS
0049           XLIN=FLOAT(I)/5.
0050           IXLIN=XLIN
0051           XLDIF=XLIN-IXLIN
0052           IF(XLDIF.GT.0.01) GOTO 140
        C     * * * SET LINE ARRAYS=DASHES * * *
0054           DO 200 JJ=1,49
0055           I1BLNK(JJ)=ICHAR1
0056    200    I2BLNK(JJ)=ICHAR1
        C     * * * SCALE WL TO FIT 50 SPACES * * *
        C      AND ROUND OFF
        C      IF WL=0 DO NOT PRINT DATA POINT
        C      PRINT 49 BLANKS OR 49 DASHES
        C      OTHERWISE INSERT AN X TO REPRESENT
        C      WL AVG FOR THAT MINUTE.
        C     * * * TEST IF WL=0 * * *
0057    140    IF(WL(I).LT.0) GOTO 150
0059           IF(IMODE.EQ.1)XDIS=(WL(I)+3)/6
0061           IF(IMODE.EQ.2)XDIS=(((VO2(I)*1000.)/WT)+.8)/1.6
0063           IXDIS=XDIS
0064           XDIF=XDIS-IXDIS
0065           IF(XDIF.GT.0.5) IXDIS=IXDIS+1
0067           J=IXDIS
0068           I1BLNK(J)=IX
        C     * * * SCALE VO2 TO FIT 49 SPACES * * *
        C     ROUNDOFF. IF VO2=0 DO NOT PRINT A DATA POINT.
        C     PRINT 49 BLANKS OR 49 DASHES.
        C     INSERT AN "X" IN THE LINE TO REPRESENT
        C      VO2 FOR THAT MINUTE.
        C     * * * TEST FOR VO2=0 * * *?
0069    150    IF(VO2(I).LT.0.01) GOTO 160
0071           XDIS1=VO2(I)*10.
0072           IXD1=XDIS1
0073           J1=IXD1
0074           I2BLNK(J1)=IX
        C     * * *  SCALE VCO2 TO FIT 49 SPACES * * *
        C      ROUNDOFF. IF VCO2=0 DO NOT PRINT A DATA POINT.
        C      PRINT 49 DASHES OR 49 BARS.
        C      INSERT AN X IN THE LINE TO REPRESENT
        C      VCO2 FOR THAT MINUTE.
        C     * * * TEST FOR VCO2=0 * * *
0075    160    IF(VCO2(I).LT.0.01) GOTO 180
0077           XDIS2=VCO2(I)*10.
0078           IXD2=XDIS2
0079           J2=IXD2
0080           I2BLNK(J2)=IO
        C     * * * TEST IF VO2=VCO2 * * *
0081           IF(J1.NE.J2) GOTO 180
0083           I2BLNK(J2)=ISTAR
        C     * * * LETS PRINT A DATA LINE * * *
0084    180    TYPE 190,ITIME(I),ILIN(I),IBAR(1),I2BLNK,
              @IBAR(1),ITIME(I),ILIN(I),IBAR(1),I1BLNK,IBAR(1)
0085    190    FORMAT(1X,A1,2X,A2,A1,49A1,A1,16X,A1,2X,
            @A2,A1,49A1,A1)
      C     * * * RESTORE LINE ARRAYS * * *
             DO 300 JJJ=1,49
             I1BLNK(JJJ)=ICHAR2
      300    I2BLNK(JJJ)=ICHAR2
      100    CONTINUE
      C     * * * NOW ADD THE BOTTOM LINE * * *
             TYPE 195,IBAR(1),I1DASH,IBAR(1),
            @IBAR(1),I1DASH,IBAR(1)
      195    FORMAT(4X,'50',A1,49A1,A1,19X,'50',
            @A1,49A1,A1)
      C     * * * OK . . . TACK ON THE SUBJ INFO * * *
             TYPE 800,CPID,CPID,SS,SS,UNG,UNG,ITDATE,IDATE
            &    ,ITDATE,IDATE
      800    FORMAT(/,5X,'CPID        :',I7,53X,
            @'CPID        :',I7,/,5X,'SS NUMBER : ',
            @I3,'-',I2,'-',I4,48X,'SS NUMBER : ',I3,'-',I2,'-',I4,
            @/,5X,'UNIQUE NO.: ',A2,I4,53X,'UNIQUE
            @ NO.: ',A2,I4,/,5X,'TEST DATE : ',I2,'/',I2,'/',I2,
            @8X,'REPORT DATE : ',2A4,A1,
            @19X,'TEST DATE : ',I2,'/',I2,'/',I2,
            @8X,'REPORT DATE : ',2A4,A1)
             TYPE 810
      810    FORMAT(20X,'NOTE: AN ASTERISK [*]
            @PLOTTED ON ANY GRAPH INDICATES BOTH
            @ VARIABLES EQUAL AT THAT MINUTE')
             RETURN
             END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

IDATE    000032  REAL*4     PARAMETER ARRAY (3)
IYLABL   000042  INTEGER*2  ARRAY (11)
IBAR     000070  INTEGER*2  ARRAY (11)
WL       000024  INTEGER*2  PARAMETER ARRAY (49)
I1DASH   000116  INTEGER*2  ARRAY (49)
I1BLNK   000260  INTEGER*2  ARRAY (49)
I2DASH   000422  INTEGER*2  ARRAY (49)
I2BLNK   000564  INTEGER*2  ARRAY (49)
ITIME    000726  INTEGER*2  ARRAY (50)
VO2      000026  REAL*4     PARAMETER ARRAY (49)
VCO2     000030  REAL*4     PARAMETER ARRAY (49)
ILIN     001072  INTEGER*2  ARRAY (49)
IFIV     001234  INTEGER*2  ARRAY (22)
IYLABX   001310  INTEGER*2  ARRAY (11)
SS       000016  INTEGER*2  PARAMETER ARRAY (3)
UNO      000020  INTEGER*2  PARAMETER ARRAY (2)
ITDATE   000022  INTEGER*2  PARAMETER ARRAY (3)
CPID     000014  INTEGER*2  PARAMETER VARIABLE
ICOPY    000034  INTEGER*2  PARAMETER VARIABLE
IMODE    000036  INTEGER*2  PARAMETER VARIABLE
WT       000040  REAL*4     PARAMETER VARIABLE
ICHAR1   001336  INTEGER*2  VARIABLE
ICHAR2   001340  INTEGER*2  VARIABLE
IC       001342  INTEGER*2  VARIABLE
IX       001344  INTEGER*2  VARIABLE
ISTAR    001346  INTEGER*2  VARIABLE
I        003002  INTEGER*2  VARIABLE
IMULT    003004  INTEGER*2  VARIABLE
XLIN     003006  REAL*4     VARIABLE
FLOAT    000000  REAL*4     PROCEDURE
IXLIN    003012  INTEGER*2  VARIABLE
XLDIF    003014  REAL*4     VARIABLE
JJ       003020  INTEGER*2  VARIABLE
XDIS     003022  REAL*4     VARIABLE
IXDIS    003026  INTEGER*2  VARIABLE
XDIF     003030  REAL*4     VARIABLE
J        003034  INTEGER*2  VARIABLE
XDIS1    003036  REAL*4     VARIABLE
IXD1     003042  INTEGER*2  VARIABLE
J1       003044  INTEGER*2  VARIABLE
XDIS2    003046  REAL*4     VARIABLE
IXD2     003052  INTEGER*2  VARIABLE
J2       003054  INTEGER*2  VARIABLE
JJJ      003056  INTEGER*2  VARIABLE
```

```
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C
          C     SUBROUTINE TO SORT THE X ARRAY DATA, AVERAGE THE CORRESPONDING
          C     Y ARRAY DATA AND THEN STORE THE AVERAGE VALUE IN THE CORRECT
          C     WORD OF THE OUT ARRAY.
          C
          C     AUTHOR :  ROY A. REED                     DATE : 27-OCT-78
          C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C
          C
0001            SUBROUTINE SORT(SFLG,IXA,IYA,IOA)
0002            INTEGER SFLG,IXA(49),IYA(49),IOA(49)
          C
          C
          D     TYPE 800,IXA,SFLG,IYA,SFLG,IOA,SFLG
          D800  FORMAT(' ###### SORT ######  ---IXA---',/,
          D    *     5(10(1X,I6)/),/,'          -----IYA-----',/,
          D    *     5(10(1X,I6)/),/,5X,'      ---IOA---',/,
          D    *     5(10(1X,I6)/))
0003            DO 100 IZ=1,49                     !LOOP TO ZERO
0004      100   IOA(IZ)=0                          ! THE OUTPUT ARRAY
          C
          C
0005            IF(SFLG.EQ.1) GO TO 300            ! TEST TO SEE IF A SORT IS NEEDED.
          C
          C
0007            DO 200 IT=1,49                     ! NO SORT NEEDED
0008      200   IOA(IT)=IYA(IT)                    ! DO A DIRECT TRANSFER OF DATA
          D     TYPE 800,IXA,SFLG,IYA,SFLG,IOA,SFLG
0009            RETURN                             ! FINSHED RETURN TO CALLING ROUTINE.
          C
          C
0010      300   DO 600 IW=1,49                     ! SORT NEEDED; SET UP LOOP TO SORT
          C                                        ! VALUES FROM SMALLEST TO LARGEST.
0011            AVG=0.                             ! ZERO THE ACCUMULATOR FOR EQUAL
          C                                        ! VALUES.
0012            ICNT=0                             ! ZERO THE COUNTER FOR EQUAL VALUES
          C                                        ! IN THE X ARRAY
          C
          C
0013            DO 500 IS=1,49                     ! LOOP TO GET ALL EQUAL VALUES IN
          C                                        ! INCREASING ORDER.
0014            IF(IXA(IS).NE.IW) GO TO 500        ! IF THE X VALUE IS GREATER THAN
          C                                        ! THE TEST VALUE BRANCH.
0016            AVG=AVG+FLOAT(IYA(IS))             ! SUMATE ALL EQUAL VALUES
0017            ICNT=ICNT+1                        ! INCREMENT THE COUNTER FOR EQUAL
          C                                        ! VALUES.
0018      500   CONTINUE
          C
          C
0019            IF (ICNT.EQ.0) GO TO 600
0021            IOA(IW)=IFIX((AVG/FLOAT(ICNT))+.5)  ! CALCULATE THE AVERAGE VALUE
0022      600   CONTINUE
```

```
FORTRAN IV      V01C-03A    FRI 29-FEB-80 14:58:40                PAGE 002

      C                                           ! AND STORE THE AVERAGE IN
      C                                           ! THE CORRECT WORD IN OUT
      C                                           ! ARRAY.
      C
0023        RETURN                         ! ALL DONE RETUN TO CALLING ROUTINE
0024        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IXA     000016  INTEGER*2 PARAMETER ARRAY (49)
IYA     000020  INTEGER*2 PARAMETER ARRAY (49)
IOA     000022  INTEGER*2 PARAMETER ARRAY (49)
SFLG    000014  INTEGER*2 PARAMETER VARIABLE
IZ      000024  INTEGER*2 VARIABLE
IT      000026  INTEGER*2 VARIABLE
IW      000030  INTEGER*2 VARIABLE
AVG     000032  REAL*4    VARIABLE
ICNT    000036  INTEGER*2 VARIABLE
IS      000040  INTEGER*2 VARIABLE
FLOAT   000000  REAL*4    PROCEDURE
IFIX    000000  INTEGER*2 PROCEDURE
```

```
         C
         C-----------------------------------------------------
0001            SUBROUTINE GRAPH(IPICK1,IPICK2,IY1,IY2,ICPID,
               *    ISS,IUNQ,ITCATE,IDATE)
         C-----------------------------------------------------------------------
         C
         C      AUTHOR:  CHARLES MANN
         C      DATE:    OCTOBER 31, 1978
         C      PURPOSE: THIS IS A GENERAL PURPOSE ROUTINE FOR PLOTTING WHATEVER
         C      DATA IS PASSED TO IT FOR THE CDAS PLOTS.
         C----------------------------------------------------------------------
         C
0002            INTEGER IY1(49),IY2(49),ISS(3),IUNQ(2),ITDATE(3)
0003            INTEGER IHEAD1(12),IHEAD2(12),IXLAB1(22),IXLAB2(22)
0004            INTEGER IYLAB1(22),IYLAB2(22),IDASH1(49),IDASH2(49),IX
0005            INTEGER IBLNK1(49),IBLNK2(49),IDASH,IBLANK
0006            REAL IDATE(3)
0007            DATA IDASH1/49*'-'/,IDASH2/49*'-'/,IBLNK1/49*' '/,IBLNK2/49*' '/
0008            DATA IX/'X'/,IDASH/'-'/,IBLANK/' '/
0009            DATA IBAR/1H\/
         D      TYPE 800,IY1,IPICK1,IY2,IPICK2
         D800   FORMAT(' ###### GRAPH ###### ',/,5(10(1X,I6),/),/,
         D     *    '                    ---- IY2 ---',/,
         D     *    5(10(1X,I6),/))
         C  PRINT CLINIC TITLE
0010            TYPE 25
0011     25     FORMAT('0',T19,'JSC CARDIOPULMONARY LABORATORY',
               *    T83,'JSC CARDIOPULMONARY LABORATORY',/)
         C  GET THE LABELS FOR THE PLOTS
0012            CALL TITLES(IPICK1,IHEAD1,IXLAB1,IYLAB1)
0013            CALL TITLES(IPICK2,IHEAD2,IXLAB2,IYLAB2)
         C  PRINT THE PLOT TITLES AND Y AXES
0014            TYPE 35,IHEAD1,IHEAD2,IYLAB1,IYLAB2,IXLAB1(1),IXLAB1(2),
               *    IDASH1,IXLAB2(1),IXLAB2(2),IDASH2
0015     35     FORMAT('0',T22,12A2,T88,12A2,/,'0',2X,11(1X,2A2),
               *    T68,11(1X,2A2),/,'  ',11(4X,'\'),T67,11(4X,'\'),/,'  ',
               *    2A2,'\',49A1,'\',T67,2A2,'\',49A1,'\')
         C-----PRINT THE PLOTS-----
0016            DO 40 LINE=1,49
0017             IF (MOD(LINE,5).NE.0) GO TO 20
         C  USE A DASHED LINE
0019            IF (IY1(LINE).LE.0) GO TO 5
0021            IF (IY1(LINE).GT.49) IY1(LINE)=49
0023            IDASH1(IY1(LINE))=IX
0024     5      IF (IY2(LINE).LE.0) GO TO 10
0026            IF (IY2(LINE).GT.49) IY2(LINE)=49
0028            IDASH2(IY2(LINE))=IX
0029     10     J=(LINE/5)*2+1
0030            TYPE 15,IXLAB1(J),IXLAB1(J+1),IDASH1,
               *    IXLAB2(J),IXLAB2(J+1),IDASH2
0031     15     FORMAT(2X,2A2,'\',49A1,'\',T67,2A2,'\',49A1,'\')
0032            IDASH1(IY1(LINE))=IDASH
0033            IDASH2(IY2(LINE))=IDASH
0034            GO TO 40
```

```
      C  USE A BLANK LINE
0035  20     IF (IY1(LINE).LE.0) GO TO 22
0037         IF (IY1(LINE).GT.49) IY1(LINE)=49
0039         IBLNK1(IY1(LINE))=IX
0040  22     IF (IY2(LINE).LE.0) GO TO 23
0042         IF (IY2(LINE).GT.49) IY2(LINE)=49
0044         IBLNK2(IY2(LINE))=IX
0045  23     TYPE 28, IBLNK1,IBLNK2
0046  28     FORMAT(6X,'\',49A1,'\',T71,'\',49A1,'\')
0047         IBLNK1(IY1(LINE))=IBLANK
0048         IBLNK2(IY2(LINE))=IBLANK
0049  40     CONTINUE
      C  PRINT THE LAST LINE AND THE SUBJECT/TEST IDENTIFICATION
      C     * * * NOW ADD THE BOTTOM LINE * * *
0050         TYPE 195,IXLAB1(21),IXLAB1(22),IBAR,IDASH1,IBAR,
            @IXLAB2(21),IXLAB2(22),IBAR,IDASH2,IBAR
0051  195    FORMAT(2X,2A2,51A1,T67,
            @2A2,51A1)
      C     * * * OK . . .  TACK ON THE SUBJ INFO * * *
0052         TYPE 50,ICPID,ICPID,ISS,ISS,IUNQ,IUNQ,ITDATE,IDATE
            &     ,ITDATE,IDATE
0053  50     FORMAT(/,5X,'CPID        :',I7,53X,
            @'CPID        :',I7,/,5X,'SS NUMBER : ',
            @I3,'-',I2,'-',I4,48X,'SS NUMBER : ',I3,'-',I2,'-',I4,
            @/,5X,'UNIQUE NO.: ',A2,I4,53X,'UNIQUE
            @ NO.: ',A2,I4,/,5X,'TEST DATE : ',I2,'/',I2,'/',I2,
            @8X,'REPORT DATE : ',2A4,A1,
            @19X,'TEST DATE : ',I2,'/',I2,'/',I2,
            @8X,'REPORT DATE : ',2A4,A1)
0054         RETURN
0055         END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IY1     000020  INTEGER*2 PARAMETER ARRAY (49)
IY2     000022  INTEGER*2 PARAMETER ARRAY (49)
ISS     000026  INTEGER*2 PARAMETER ARRAY (3)
IUNG    000030  INTEGER*2 PARAMETER ARRAY (2)
ITDATE  000032  INTEGER*2 PARAMETER ARRAY (3)
IHEAD1  000036  INTEGER*2 ARRAY (12)
IHEAD2  000066  INTEGER*2 ARRAY (12)
IXLAB1  000116  INTEGER*2 ARRAY (22)
IXLAB2  000172  INTEGER*2 ARRAY (22)
IYLAB1  000246  INTEGER*2 ARRAY (22)
IYLAB2  000322  INTEGER*2 ARRAY (22)
IDASH1  000376  INTEGER*2 ARRAY (49)
IDASH2  000540  INTEGER*2 ARRAY (49)
IBLNK1  000702  INTEGER*2 ARRAY (49)
IBLNK2  001044  INTEGER*2 ARRAY (49)
IDATE   000034  REAL*4    PARAMETER ARRAY (3)
IPICK1  000014  INTEGER*2 PARAMETER VARIABLE
IPICK2  000016  INTEGER*2 PARAMETER VARIABLE
ICPID   000024  INTEGER*2 PARAMETER VARIABLE
IX      001206  INTEGER*2 VARIABLE
IDASH   001210  INTEGER*2 VARIABLE
IBLANK  001212  INTEGER*2 VARIABLE
IBAR    001214  INTEGER*2 VARIABLE
TITLES  000000  REAL*4    PROCEDURE
LINE    002134  INTEGER*2 VARIABLE
MOD     000000  INTEGER*2 PROCEDURE
J       002136  INTEGER*2 VARIABLE
```

```
      C---------------------------------------------------------
0001        SUBROUTINE TITLES(ICODE,IHEAD,IXLABL,IYLABL)
      C------------------------------------------------------------
      C
      C     AUTHOR:  CHARLES MANN
      C     DATE    OCTOBER 31, 1978
      C     PURPOSE: TO STORE THE TITLE AND AXIS LABELS IN
      C     THE APPROPRIATE ARRAYS
      C-------------------------------------------------------------
      C
0002        INTEGER IHEAD1(12,11),IXLABL(22),IYLABL(22),IYXVAL(2,11)
0003        INTEGER ILABEL(22,5),IHEAC(12)
0004        DATA IHEAD1/
           1    '  ','  ','MI','N ','VO','L ','VS',' T','IM','E ','  ','  ',
           2    '  ',' R','ES','P ','RA','TE',' V','S ','TI','ME','  ','  ',
           3    'VO','2 ','VS',' T','IM','E ','(M','L/','KG','-M','IN',') ',
           4    '  ','  ','  ',' V','C2',' V','S ','WL','  ','  ','  ','  ',
           5    '  ','  ','  ','  ','HR',' V','S ','WL','  ','  ','  ','  ',
           6    '  ','  ','  ',' S','BP',' V','S ','WL','  ','  ','  ','  ',
           7    '  ','  ','  ',' S','BP',' V','S ','HR','  ','  ','  ','  ',
           8    '  ','  ','  ',' S','BP',' V','S ','VO','2 ','  ','  ','  ',
           9    '  ','  ',' V','CO','2 ','VS',' V','O2','  ','  ','  ','  ',
           A    '  ','  ',' M','IN',' V','OL',' V','S ','VO','2 ','  ','  ',
           B    '  ','  ','MI','N ','VO','L ','VS',' V','CO','2 ','  ','  '/
0005        DATA ILABEL/
           1    '  ',' 0','  ',' 5','  ','10','  ','15','  ','20','  ','25',
           &    '  ','30','  ','35','  ','40','  ','45','  ','50',
           2    '  ',' 0','  ','20','  ','40','  ','60','  ','80',' 1','00',
           &    ' 1','20',' 1','40',' 1','60',' 1','80',' 2','00',
           3    '  ',' 0','  ','30','  ','60','  ','90',' 1','20',' 1','50',
           &    ' 1','80',' 2','10',' 2','40',' 2','70',' 3','00',
           4    ' 0','.0',' 0','.5',' 1','.0',' 1','.5',' 2','.0',' 2','.5',
           &    ' 3','.0',' 3','.5',' 4','.0',' 4','.5',' 5','.0',
           5  '  ',' 0','  ','10','  ','20','  ','30','  ','40','  ','50',
           &  '  ','60','  ','70','  ','80','  ','90',' 1','00'/
0006        DATA IYXVAL/2,1,    5,1,    1,1,    4,3,    2,3,    3,3,
           *             3,2,    3,4,    4,4,    2,4,    2,4/
      D     TYPE 800,ICODE
      D800  FORMAT(' ###### TITLES ######  ICCDE=',I3)
0007        JCODE=ICODE-1
0008        DO 100 I=1,12
0009  100   IHEAD(I)=IHEAD1(I,JCODE)
0010        DO 200 I=1,22
0011         IXLABL(I)=ILABEL(I,IYXVAL(2,JCODE))
0012         IYLABL(I)=ILABEL(I,IYXVAL(1,JCCDE))
0013  200   CONTINUE
0014        RETURN
0015        END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IHEAD1  000024  INTEGER*2 ARRAY (12,11) VECTORED
IXLABL  000020  INTEGER*2 PARAMETER ARRAY (22)
IYLABL  000022  INTEGER*2 PARAMETER ARRAY (22)
IYXVAL  000434  INTEGER*2 ARRAY (2,11)
ILABEL  000510  INTEGER*2 ARRAY (22,5) VECTORED
IHEAD   000016  INTEGER*2 PARAMETER ARRAY (12)
ICODE   000014  INTEGER*2 PARAMETER VARIABLE
JCODE   001104  INTEGER*2 VARIABLE
I       001106  INTEGER*2 VARIABLE
```

```
C   ********************************************************************
C   ********************************************************************
C   *                                                                  *
C   *                   TECHNOLOGY INCORPORATED                        *
C   *                    LIFE SCIENCE DIVISION                         *
C   *                                                                  *
C   ********************************************************************
C   *                                                                  *
C   *   PROGRAM NAME:................. EDICON                          *
C   *   AUTHOR:....................... ROY A. REED                     *
C   *   DATE:......................... 2/NOV/78                        *
C   *                                                                  *
C   *------------------------------------------------------------------*
C   *                                                                  *
C   *   COMPUTER SYSTEM:............. DEC PDP 1103 "LSI-11"            *
C   *   OPERATING SYSTEM:............. RT-11  V01C-03A                 *
C   *                                                                  *
C   *------------------------------------------------------------------*
C   *                                                                  *
C   *   COMPILING SEQUENCE:                                            *
C   *                                                                  *
C   *      .R FORTRA <CR>                                              *
C   *      *EDICON=EDICON <CR>                                         *
C   *      *<CR>                                                       *
C   *                                                                  *
C   *------------------------------------------------------------------*
C   *                                                                  *
C   *   RUN MODUAL LINKING SEQUENCE:                                   *
C   *                                                                  *
C   *      .R LINK <CR>                                                *
C   *      *EDICON=EDICON,DX0:SYSLIB/F <CR>                            *
C   *      *<CR>                                                       *
C   *                                                                  *
C   *------------------------------------------------------------------*
C   *                                                                  *
C   *   CALLING SEQUENCE:                                              *
C   *                                                                  *
C   *      .R EDICON <CR>                                              *
C   *                                                                  *
C   *------------------------------------------------------------------*
C   *                                                                  *
C   *   PURPOSE:                                                       *
C   *                                                                  *
C   *      ALLOWS THE USER TO MODIFY THE CONSTANTS USED BY THE         *
C   *      CALIBRATION ROUTINE IN THE MAIN PROGRAM PROCES.             *
C   *      THE CONSTANTS ARE STORED IN THE DISK FILE CALCON.DAT.       *
C   *                                                                  *
C   ********************************************************************
C   ********************************************************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
FORTRAN IV      V01C-03A   FRI 29-FEB-80 14:03:01               PAGE 001

0001            LOGICAL*1 IANS,YES,NO
0002            INTEGER FILE(4)
0003            REAL CALBUF(256)
0004            DATA YES/1HY/,NO/1HN/
0005            CALL IRAD50(12,'DXOCALCONCAT',FILE)
0006            ICHAN=IGETC()
0007            IF(LOOKUP(ICHAN,FILE).LT.0) STOP 'LOOKUP ERROR'
0009            IF(IREADW(256,CALBUF,1,ICHAN).LT.0) STOP 'DISK READ ERROR'
0011            TYPE 2
0012      2     FORMAT(/,'$  LIST OF CURRENT VALUES NEEDED  ?? YES OR NO ??')
0013            ACCEPT 3,IANS
0014      3     FORMAT(A1)
0015            IF(IANS.NE.YES)GO TO 8
0017            DO 5 I=1,22
0018            TYPE 4,I,CALBUF(I)
0019      4     FORMAT(/,1X,'CALIBRATION FOR WORD ',I2,' IS ',F10.2)
0020      5     CONTINUE
0021      8     TYPE 6
0022      6     FORMAT(/,'$ TYPE IN NUMBER OF WORD TO EDIT ')
0023            ACCEPT 7,I
0024      7     FORMAT(I3)
0025            IF(I.EQ.0)GO TO 20
0027            TYPE 10,I,CALBUF(I)
0028     10     FORMAT(/,'$  WORD ',I4,' )   OLD = ',F10.2,'     NEW = ',)
0029            ACCEPT 15,CALBUF(I)
0030     15     FORMAT(F10.0)
0031            GO TO 8
0032     20     CONTINUE
0033            IF(IWRITW(256,CALBUF,1,ICHAN).LT.0)STOP 'DISK WRITE ERROR'
0035            CALL  CLOSEC(ICHAN)
0036            STOP
0037            END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

FILE     000006  INTEGER*2 ARRAY (4)
CALBUF   000016  REAL*4    ARRAY (256)
IANS     002412  LOGICAL*1 VARIABLE
YES      002016  LOGICAL*1 VARIABLE
NO       002017  LOGICAL*1 VARIABLE
IRAD50   000000  INTEGER*2 PROCEDURE
ICHAN    002414  INTEGER*2 VARIABLE
IGETC    000000  INTEGER*2 PROCEDURE
LOOKUP   000000  INTEGER*2 PROCEDURE
IREADW   000000  INTEGER*2 PROCEDURE
I        002416  INTEGER*2 VARIABLE
IWRITW   000000  INTEGER*2 PROCEDURE
CLOSEC   000000  REAL*4    PROCEDURE
```

APPENDIX I

FORMAT OF THE FLOPPY DISK FILE

CDAS PATIENT DATA FILE - FIRST BLOCK - PRETEST DATA

DUM 1 1	SUBJECT SEX A 2	SOCIAL I 3	SECURITY I 4	NUMBER I 5	WLSUM I 6	UNIQUE A 7	NUMBER I 8	RETEST I 9	CPID I 10	TEST MODE 1=LIVE 2=TAPE I 11	TYPE OF TEST B=1,T=2 I 12

MAJOR REV. # I 13	MINOR REV. # I 14	TARGET HR @4 I 15	TARGET HR @8 I 16	TARGET HR @12 I 17	TARGET HR @16 I 18	DUM 4 19	MAX TARGET HR I 20	DUM 5 21	DATE OF MONTH I 22	BIRTH DAY I 23	YEAR I 24

AGE I 25	SUBJECT NAME (14 WORDS, 28 CHARACTERS) → A 26	A 27	A 28	A 29	A 30	A 31	A 32	A 33	A 34	A 35	A 36

SUBJECT NAME CONT'D A 37	A 38	A 39	TEST DATE MONTH I 40	DAY I 41	YEAR I 42	CALIBRATION FLAG 43	DUM 6 (6 WORDS) → 44	45	46	47	48

DUM 6 49	HEIGHT (CM) → F 50	F 51	WEIGHT (KG) → F 52	F 53	AMBIENT TEMP. (°C) → F 54	F 55	DUM 7 56	AMBIENT PRES. (mm Hg) → F 57	F 58	DUM 8 (2 WORDS) 59	60

TIME BEGIN REST HH I 61	MM I 62	SS I 63	NO.OF REST RECORDS I 64	TIME BEGIN EXERCISE HH I 65	MM I 66	SS I 67	NO.OF EXERCISE RECORDS I 68	TIME BEGIN RECOVERY HH I 69	MM I 70	SS I 71	NO.OF RECOVERY RECORDS I 72

FIRST BLOCK (CONTINUED) - CALIBRATION & PFT DATA

STPD FACTOR		BTPS	FACTOR	AMBIENT	O_2%	AMBIENT	N_2%	AMBIENT	CO_2%	DUM 9 (	2 WORDS)
F 73	F 74	F 75	F 76	F 77	F 78	F 79	F 80	F 81	F 82	83	84

SP. VOL.	SLOPE	SP. VOL.	Y-INT	O_2 SLOPE		O_2 Y-INT		N_2 SLOPE		N_2 Y-INT	
F 85	F 86	F 87	F 88	F 89	F 90	F 91	F 92	F 93	F 94	F 95	F 96

CO_2 SLOPE		CO_2 Y-INT		HR-SLOPE		HR-Y-INT		WL/ELEV	SLOPE	WL/ELEV	Y-INT
F 97	F 98	F 99	F 100	F 101	F 102	F 103	F 104	F 105	F 106	F 107	F 108

SPEED	SLOPE	SPEED	Y-INT	SBP	SLOPE	SBP Y-INT		DBP SLOPE		DBP Y-INT	
F 109	F 110	F 111	F 112	F 113	F 114	F 115	F 116	F 117	F 118	F 119	F 120

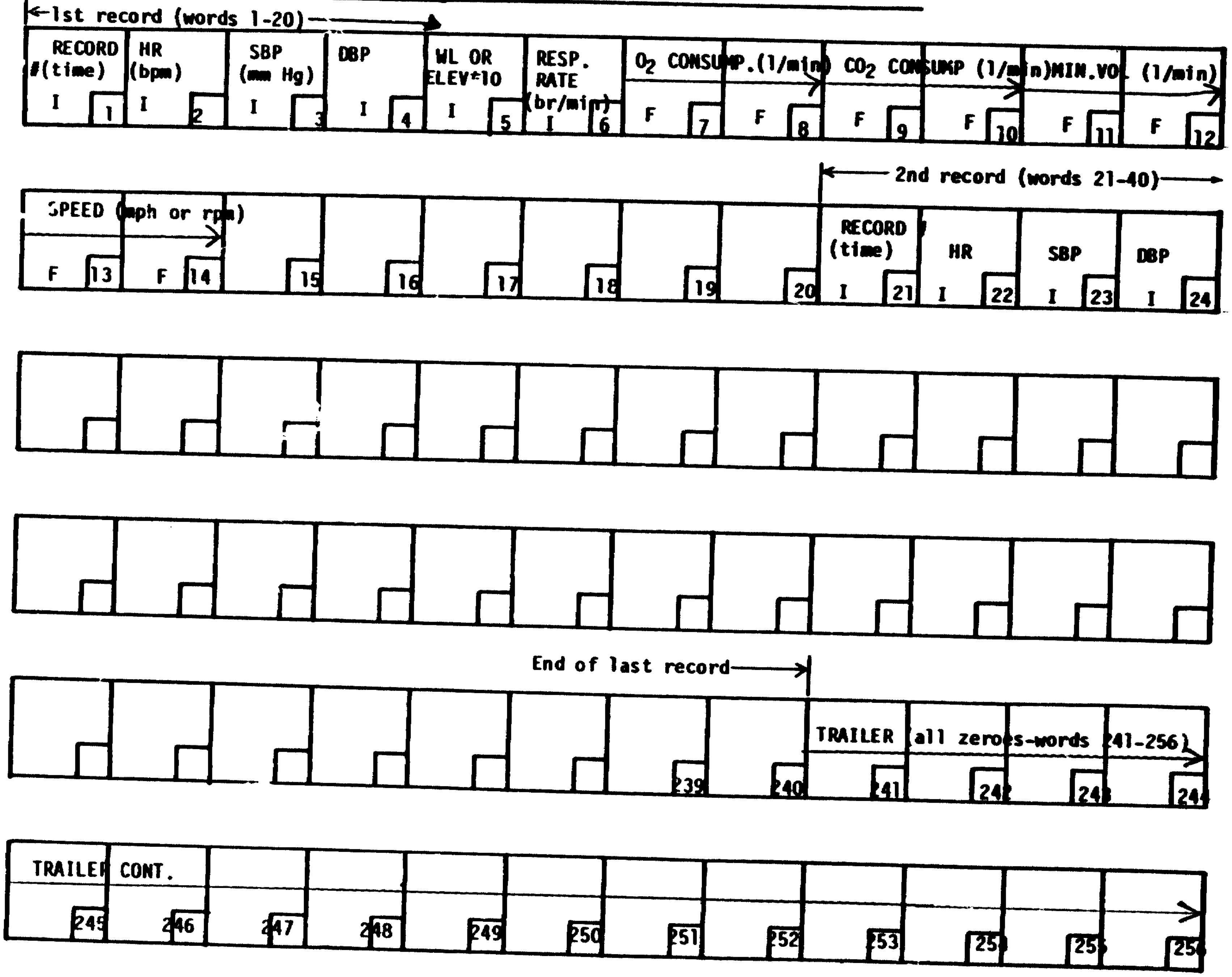
								FVC		FEV1	
121	122	123	124	125	126	127	128	129	130	131	132

PEFR		MMFR		MEFR				% FEV1/FVC			
F 133	F 134	F 135	F 136	F 137	F 138	139	140	F 141	F 142	F 143	F 144

FIRST BLOCK (CONTINUED) - PFT DATA

PRED FVC		% OBS/PRED FVC		PRED FEV1		% OBS/PRED FEV1					
F 145	F 146	F 147	F 148	F 149	F 150	F 151	F 152	153	154	155	156

CDAS PATIENT DATA FILE - LAST FIVE BLOCKS - EXERCISE DATA



* U.S. GOVERNMENT PRINTING OFFICE: 1980 - 671-099/689